EXHIBIT A

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SALESFORCE.COM, INC.
Petitioner

v.

WSOU INVESTMENTS, LLC d/b/a
BRAZOS LICENSING AND DEVELOPMENT
Patent Owner

U.S. Patent No. 8,280,928

"Multi-Level Enmeshed Directory Structures"

Inter Partes Review No. IPR2022-00428

PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 8,280,928 UNDER 35 U.S.C. §§ 311–319 AND 37 C.F.R. §§ 42.100 *ET SEQ.*

TABLE OF CONTENTS

Page

LIST OF EXHIBITS

Exhibit	Description
Ex. 1001	U.S. Patent No. 8,280,928 ("the '928 patent")
Ex. 1002	File History of U.S. Patent No. 8,280,928 (U.S. Patent Application No. 12/415,375)
Ex. 1003	Declaration of Dr. Douglas C. Schmidt ("Schmidt")
Ex. 1004	Kermal Erdogan, *A Model to Represent Directed Acyclic Graphs (DAG) on SQL Databases*, Code Project (Jan. 14, 2008), https://www.codeproject.com/script/Articles/ArticleVersion.aspx?aid=22824&av=37722 ("Erdogan")
Ex. 1005	U.S. Patent No. 9,104,779 ("Hunt")
Ex. 1006	U.S. Patent No. 5,644,686 ("Hekmatpour")
Ex. 1007	U.S. Patent No. 8,195,646 ("Evans")
Ex. 1008	U.S. Patent Application Publication No. 2008/0279273 ("Liang")
Ex. 1009	U.S. Patent Application Publication No. 2004/0148296 ("Schaepe")
Ex. 1010	Curriculum Vitae of Dr. Douglas C. Schmidt
Ex. 1011	Scheduling Order (Nov. 29, 2021) (Dkt. 44)
Ex. 1012	Email with Mr. Gunnell Regarding Case Schedule (Sept. 9, 2021)
Ex. 1013	Email with Mr. Gunnell Regarding Case Schedule (Nov. 30, 2021)
Ex. 1014	Email with Mr. Gunnell Regarding Case Schedule (Dec. 13, 2021)
Ex. 1015	Summons in a Civil Action (Feb. 24, 2021) (Dkt. 12)
Ex. 1016	Return of Service (Mar. 2, 2021) (Dkt. 14)
Ex. 1017	Parties' Preliminary Claim Constructions
Ex. 1018	Salesforce's Opening Claim Construction Brief (Nov. 1, 2021) (Dkt. 36)
Ex. 1019	Plaintiff's Responsive Claim Construction Brief (Nov. 3, 2021) (Dkt. 38)

Exhibit	Description
Ex. 1020	Salesforce's Reply Claim Construction Brief (Nov. 18, 2021) (Dkt. 39)
Ex. 1021	Plaintiff's Sur-Reply Claim Construction Brief (Dec. 1, 2021) (Dkt. 45)
Ex. 1022	Original Complaint for Patent Infringement (Dec. 18, 2020) (Dkt. 1)
Ex. 1023	Declaration of Dr. Sylvia D. Hall-Ellis
Ex. 1024	Curriculum Vitae of Dr. Sylvia D. Hall-Ellis

LIST OF CHALLENGED CLAIMS

Claim 1	
1pre	A method for generating a multi-level hierarchical directory structure and establishing relationships between descriptors, the method comprising:
1a	selecting an initial data object;
1b	creating one or more descriptors associated with the data object wherein each of said descriptors are further associated with one or more corresponding descriptors thereby forming a multi-level relational tree;
1c	determining the relationship between the one or more descriptors;
1d	creating a hierarchical structure linking the different levels of descriptors;
1e	updating a corresponding database; and
1f	identifying a single initial descriptor that links a plurality of descriptors and two or more predecessor descriptors linking another single descriptor to thereby establish relationships between different descriptors relative to themselves and to the single initial descriptor.
Claim 2	
The method of claim 1, wherein each descriptor can be related to one or more predecessor descriptors thereby forming a hierarchical relationship.	

Claim 3	
The method of claim 1, wherein each descriptor can be associated with one or more predecessor descriptors and the relationship of the object to the one or more predecessor descriptors is acyclic.	
Claim 4	
The method of claim 1, further comprising determining the relationships between different descriptors relative to themselves and to the initial data object.	
Claim 5	
The method of claim 1, wherein the operation further comprises a graphical user interface (GUI) to navigate the enmeshed directory in both directions, said GUI presents both descriptors or objects described by a particular descriptor using proper links and the descriptors describing the object or descriptor using proper links.	
Claim 6	
6pre	A network management system communicatively coupled to one or more element management systems adapted to perform a method for creating a multi-level hierarchical directory structure and establishing relationships between descriptors, comprising:
6a	a processor for executing software instructions received from a memory to perform thereby a method for, the method comprising:
6b	linking each of a plurality of data objects to multiple respective descriptors, each of said descriptors being linked with one or more predecessor tags; and
6c	identifying a single initial descriptor that links a plurality of descriptors and two or more predecessor descriptors linking another single descriptor to thereby establish the relationships between different descriptors relative to themselves and to the single initial descriptor.

Claim 7	
The system of claim 6, wherein the operation further comprises a graphical user interface (GUI) to navigate the enmeshed directory in both directions, said GUI presents both descriptors or objects described by a particular descriptor using proper links and the descriptors describing the object or descriptor using proper links.	
Claim 8	
The method of claim 6, wherein the relationships of the object to the one or more descriptors is acyclic.	
Claim 13	
13pre	A computer readable storage medium for storing instructions which, when executed by one or more processors communicatively coupled to a network, perform a method for creating a multi-level hierarchical directory structure and establishing relationships between descriptors, comprising:
13a	linking by a device an object to multiple descriptors describing said object, each of said descriptors being identified by one or more predecessor descriptors linked to the descriptor; and
13b	identifying a single initial descriptor that links a plurality of descriptors and two or more predecessor descriptors linking another single descriptor to thereby establish relationships between different descriptors relative to themselves and to the single initial descriptor.
Claim 14	
The computer readable medium of claim 13, wherein navigating the enmeshed directory in both directions requires the GUI, said GUI presents both descriptors or objects described by a particular descriptor using proper links and the descriptors describing the object or descriptor using proper links.	
Claim 15	
The computer readable medium of claim 13, wherein the relationship of the object to the one or more descriptors is acyclic.	

Claim 16	
The computer readable medium of claim 13, further comprising determining the relationships between different descriptors relative to themselves and to the initial data object.	
Claim 17	
17pre	A content server comprising a processor, said content server multicasting to a plurality of client servers in a network system adapted to perform a method for creating a multi-level hierarchical directory and establishing relationships between descriptors, the method comprising:
17a	linking by the content server each of a plurality of client servers to multiple respective descriptors, each of said descriptors being linked with one or more predecessor descriptors wherein a top level predecessor descriptor corresponds to the content server; and
17b	identifying a single initial descriptor that links a plurality of client servers and two or more predecessor descriptors linking another single descriptor to thereby establish relationships between different descriptors relative to themselves and to the single initial client server.
Claim 18	
The content server of claim 17, further comprising generating a graphical user interface (GUI) to visualize navigation of the enmeshed directory in both directions, said GUI presenting both descriptors or clients servers described by a particular descriptor using proper links and descriptors describing the client server or descriptor using proper links.	
Claim 19	
The network system of claim 17, wherein each descriptor can be associated with one or more predecessor descriptors and the relationships of the client server to the one or more descriptors is acyclic.	

I. INTRODUCTION

Patent Owner Brazos Licensing and Development ("Brazos") is a non-practicing entity, and states that it is based in Waco, Texas. Brazos recently sued Petitioner Salesforce simultaneously filing nine other litigations alleging infringement of nine separate patents. All of these patents stem from different patent families.

One of those ten patents, U.S. Patent No. 8,280,928 ("'928 patent") is the subject of the present petition. The '928 patent is directed to creating a multi-level directory structure for storing and organizing data objects. Ex. 1001 at Abstract. The '928 patent purports to invent a directory structure where the elements at the bottom level ("data objects") can be linked to multiple elements at higher levels ("descriptors"), and the descriptors can in turn be linked to each other. Ex-1001 at 1:43–44. This structure, however, was well known at the time of filing the '928 patent and is rendered obvious by the prior art teachings detailed below.

Accordingly, Petitioner salesforce.com, inc. ("Salesforce" or "Petitioner") respectfully requests *inter partes* review of claims 1–8 and 13–19 ("the challenged claims") of the '928 patent assigned to WSOU Investments, LLC ("Patent Owner" or "PO"), pursuant to 35 U.S.C. § 311 and 37 C.F.R. § 42.100.

II. MANDATORY NOTICES

Pursuant 37 C.F.R. § 42.8(a)(1), the following mandatory notices are provided

as part of this Petition.

A. Real Party-in-Interest (37 C.F.R. § 42.8(b)(1))

The real party-in-interest for Petitioner is Salesforce.

B. Related Matters (37 C.F.R. § 42.8(b)(2))

1. Related Patent Office Proceedings

No pending U.S. patent application claims a filing benefit from U.S. Application No. 12/415,375, which was granted as the '928 patent.

2. Related Litigation

The '928 patent is the subject of co-pending district court litigation: *WSOU Investments, LLC d/b/a Brazos Licensing and Development v. Salesforce.com, Inc.*, Case No. 6:20-cv-01163-ADA (W.D. Tex.) ("the Litigation"). PO asserted claims 1–5 and 13–16 in the district court litigation.

To Petitioner's knowledge, the '928 patent has not been the subject of any other district court or PTAB litigation.

3. Related Applications

The '928 patent does not claim priority to any other application.

C. Lead and Back-Up Counsel (37 C.F.R. § 42.8(b)(3)) and Service Information (37 C.F.R. § 42.8(b)(3)–(4))

Petitioner provides the following counsel and service information.[1] Pursuant to 37 C.F.R. § 42.10(b), a Power of Attorney accompanies this Petition.

LEAD COUNSEL	BACK-UP COUNSEL
James Glass Reg. No. 46729 jimglass@quinnemanuel.com Quinn Emanuel Urquhart & Sullivan, LLP 51 Madison Ave., 22nd Floor New York, NY 10010 Tel: (212) 849-7000	Ray Zado (Back-up Counsel) (*pro hac vice* to be requested upon grant authorization) rayzado@quinnemanuel.com Quinn Emanuel Urquhart & Sullivan, LLP 555 Twin Dolphin Dr., 5th Floor Redwood Shores, CA 94065 Tel: (650) 801-5000 Sam Stake (Back-up Counsel) (*pro hac vice* to be requested upon grant authorization) samstake@quinnemanuel.com Quinn Emanuel Urquhart & Sullivan, LLP 50 California St., 22nd Floor San Francisco, CA 94111 Tel: (415) 875-6600 Jared Kneitel (Back-up Counsel) Reg. No. 51178 jaredkneitel@quinnemanuel.com

[1] Petitioner consents to electronic service to qe-kaufman-salesforce@quinnemanuel.com and the email addresses listed in the table below.

	Quinn Emanuel Urquhart & Sullivan, LLP 1109 First Avenue Suite 210 Seattle, WA 98101 Tel: (206) 905-7000 Joshua Scheufler (Back-up Counsel) (*pro hac vice* to be requested upon grant authorization) joshuascheufler@quinnemanuel.com Quinn Emanuel Urquhart & Sullivan, LLP 711 Louisiana Street, Suite 500 Houston, TX 77002 Tel: (713) 221-7000

Petitioner consents to electronic service at the email addresses listed above.

D. Payment of Fees (37 C.F.R. § 42.15(a))

The undersigned authorizes the Office to charge the fee required for this Petition for *inter partes* review to Deposit Account No. 50-5708. Any additional fees that might be due are also authorized.

III. REQUIREMENTS FOR INTER PARTES REVIEW

Petitioner certifies it is not barred or estopped from requesting this proceeding, the ’928 patent is available for *inter partes* review, and the prohibitions of 35 U.S.C. §§ 315(a)–(b) are inapplicable.

IV. STATEMENT OF RELIEF REQUESTED FOR EACH CHALLENGED CLAIM

Petitioner respectfully requests that *inter partes* review of the challenged claims be instituted based on the following grounds:

#	Ground for Challenge
1	Erdogan and Evans render claims 1–4 obvious
2	Erdogan, Evans, and Hekmatpour render claim 5 obvious
3	Erdogan, Evans, Hekmatpour, and Hunt render claims 6–8, 13–19 obvious

V. OVERVIEW OF THE ’928 PATENT

A. Technology

The ’928 patent relates to a method for creating a multi-level enmeshed structure. Ex-1001 at Abstract. In contrast, a “simple” directory structure typically associates a data object with a unique descriptor—such as a filename—which can then be associated with a unique chain of higher-level descriptors—such as a file path. *Id.* at 1:31–34. In such a structure, there is a unique chain of ordered descriptors leading to each object. *Id.* at 1:36–37; Ex. 1003 ¶ 28.

The ’928 patent purports to introduce a novel structure for descriptors. In what it refers to as an “enmeshed directory structure,” multiple descriptors may describe a single data object; each descriptor may in turn also have multiple

descriptors; and the descriptors may be linked to each other across levels. Ex. 1001 at 1:43–44.

> 1. A method for generating a multi-level hierarchical directory structure and establishing relationships between descriptors, the method comprising:
>
> selecting an initial data object;
>
> creating one or more descriptors associated with the data object wherein each of said descriptors are further associated with one or more corresponding descriptors thereby forming a multi-level relational tree;
>
> determining the relationship between the one or more descriptors;
>
> creating a hierarchical structure linking the different levels of descriptors;
>
> updating a corresponding database; and
>
> identifying a single initial descriptor that links a plurality of descriptors and two or more predecessor descriptors linking another single descriptor to thereby establish relationships between different descriptors relative to themselves and to the single initial descriptor.

This claimed multi-level data structure, however, was well-known in the prior art, including the Erdogan, Hunt, Hekmatpour, and Evans references. Ex. 1003 ¶ 29.

B. Priority Date

Because the prior art cited in this Petition predates the earliest possible priority date of the ’928 patent (March 31, 2009) by more than one year, each cited reference herein qualifies as prior art under at least one of 35 U.S.C. 102(a), (b), and/or (e).

VI. PERSON OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art ("POSITA") relating to the subject matter of the '928 patent is a person with a Bachelor of Science degree in electrical engineering or computer science or related field plus at least two years of professional experience in the field of relational databases and/or user interfaces thereto. Ex. 1003 ¶ 36.

VII. CLAIM CONSTRUCTION

The construction process has begun in the litigation—the parties have exchanged proposed constructions, Ex. 1017, and finished briefing, Ex. 1018–1021. A *Markman* hearing for the '928 patent will be scheduled for early January of 2022. Ex. 1014. The only two terms in contest between the parties regarding the '928 patent are (1) "directory" and (2) "identifying a single initial descriptor that links a plurality of descriptors and two or more predecessor descriptors linking another single descriptor" in claims 1 and 13. Below are the parties competing constructions:

Salesforce's Proposed Construction	WSOU's Proposed Construction
"directory"	
"An entity in a file system which contains a group of files and/or other directories"	This Preamble term is not limiting
"identifying a single initial descriptor that links a plurality of descriptors and two or more predecessor descriptors linking another single descriptor"	

"identifying a single initial descriptor in a first level of the hierarchy, the single initial descriptor linked to a plurality of descriptors in a second level of the hierarchy, the plurality of descriptors linked to two or more predecessor descriptors in a third level of the hierarchy, and the two or more predecessor descriptors linked to another single initial descriptor that is in a fourth level of the hierarchy"	Plain and ordinary meaning

Petitioner's construction is correct as it most naturally aligns with the intrinsic evidence, including the claim language, the specification, and the prosecution history.

The "directory" term as it appears in the preamble is limiting because the term "recites an essential structure to the very concept of the invention" and "serves as antecedent basis for a number of limitations in the independent and dependent claims." Ex. 1018 at 11. Petitioner's construction derives directly from the patentee's lexicography, where the patentee defines the term "directory" in the Background of the invention section as follows: "[A] directory ***is*** an entity in a file system, which contains a group of files and/or other directories." *Id.* at 11 (quoting Ex. 1001 at 1:25–27) (emphasis added). Such use of the term "***is***" is the quintessential mechanism by which a patentee act as his own lexicographer. *Id.* at 11–12.

For the "identifying …" term, the claim language itself reflects that the resultant claimed directory structure is a "multi-level hierarchical" structure comprising four different types of descriptors: "a single initial descriptor," "a plurality of descriptors," "two or more predecessor descriptors," and "another single descriptor," which are all linked in a particular way. *Id.* at 12.

The specification likewise described a four level hierarchy in accordance with the statements in the file history. It specifically refers to the claimed four levels of the hierarchy:

> Specifically, each of a plurality of objects (collectively objects 310) is linked to one or more of eight descriptors (A5-A8; B4-B7) at a first hierarchical level, denoted as level 320. Each of the descriptors at the first hierarchical level 320 is linked to one or more of six descriptors (A2-A4; B2-B3) at a second hierarchical level 330. Each of the descriptors at the second article level 330 is linked to one or more of two descriptors (A1; B1) at a third hierarchical level 340.

These first, second, and third hierarchical levels in the specification correspond to the claimed second, third, and fourth hierarchical levels, with a plurality of objects (collectively objects 310) being on the first hierarchical level. *Id.* at 14–15.

Also, critically, Petitioner's construction of the "identifying …" limitation follows from the prosecution history, where the applicant made arguments to overcome the prior art and, in so doing, expressly characterized the claimed invention to distinguish that art. In particular, the applicant filed an appeal brief, in which applicant expressly characterized this claim limitation, arguing based on a

direct mapping to this limitation that "***[a]s claimed, there are at least four levels to the hierarchy***; namely, ([level] 1) an initial descriptor . . . that ([level] 2) links a plurality of descriptors . . . and ([level] 3) two or more predecessor descriptors . . . linking ([level] 4) another single descriptor." *Id.* at 13.

In contrast, WSOU suggests that these two terms simply be given its "plain and ordinary meaning," without specifying precisely, the scope of its proposed "plain and ordinary meaning." *See generally* Ex. 1019 at 11–15. For the "directory" term, WSOU simply argues that the term is not limiting, and that "[n]either claim [claims 1 and 13] has been shown to affirmatively require a 'directory' itself." *Id.* at 11. For the "identifying …" term, WSOU made plain that its interpretation of the terms does not require the four level hierarchical structure embodied within Petitioner's construction.[2]

[2] First, WSOU suggests that "the single initial descriptor" need not be "in a first level of the hierarchy" and "linked to a plurality of descriptors in a second level of the hierarchy." *See id.* at 14. Second, it argues that "[t]here is no such unambiguous *located* "*in*" requirement for the descriptors themselves." *Id.* Third, WSOU believes that "another single initial descriptor" should not be limited to being located "in" an unrecited "fourth level of the hierarchy." *Id.* at 15.

While WSOU's construction of either term is plainly contrary to the claims, specification, and prosecution history, resolution of the dispute over the breadth of the terms is unnecessary for purposes of this *inter partes* review. This Petition demonstrates that all claims are invalid under either construction. The grounds set forth in this Petition address Petitioner's narrower constructions. Therefore, even if the Board were to construe both terms according to WSOU's constructions, the Petition would only demonstrate the prior art discloses additional limitations. Ex. 1003 ¶ 38.

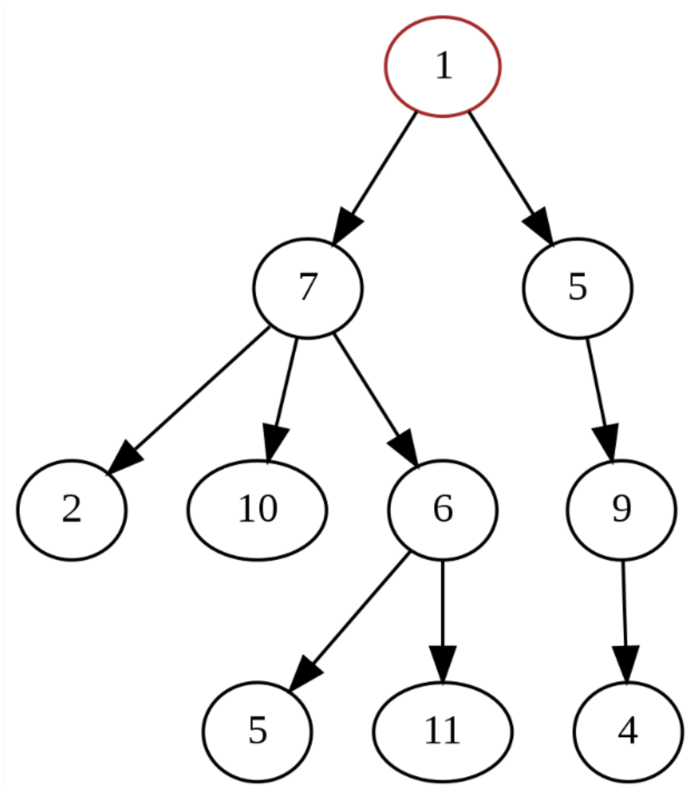
VIII. ASSERTED PRIOR ART

A. Technologies

All the prior art references discussed herein are in the same field of endeavor as the '928 patent, i.e., representing and generating hierarchical relationships in computer file systems, databases, expert systems, or knowledge representations. *See* Ex-1001 at 1:58–2:2; Ex-1004 at Introduction; Ex-1005 at 1:54–57; Ex-1006 at Abstract; Ex-1007 at Abstract. Thus, it is helpful to first explain some of the common terminologies shared by these references.

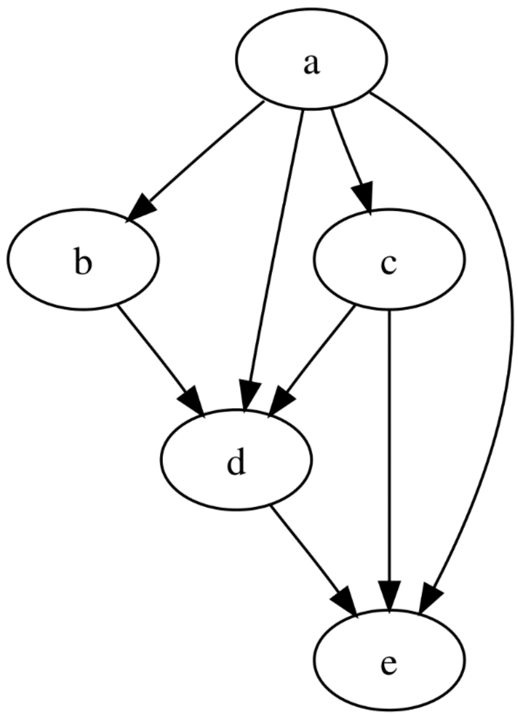
In this field, a tree is a widely used abstract data type that simulates a hierarchical data structures, with a root value and subtrees of children with a parent node, represented as a set of linked nodes. In the diagram below, the node labeled 7

has three children, labeled 2, 10 and 6, and one parent, labeled 1. The root node, at the top, has no parent. Ex. 1003 ¶ 45.



A tree is a more restricted type of the so-called Directed Acyclic Graphs (DAGs). Unlike trees, which can only have one parent, a DAG can have multiple parents. The defining feature of a DAG is that it consists of vertices and edges, with each edge directed from one vertex to another, such that following those directions will never form a closed loop. Ex-1004 at Introduction. The diagram below is an example of a DAG. The nodes represented by a, b, c, d, and e are referred to as ***vertices***. The arrows connecting the vertices are referred to as ***edges***. Because the

edges have directions, the diagram is said to be "***directed***." Finally, because following those directions will never form a closed loop, the diagram is also said to be "***acyclic***." Ex. 1003 ¶¶ 46–48.



B. Erdogan

Erdogan, published on January 14, 2008,[3] is directed to representing hierarchical relationships in data sets, i.e., a hierarchical relational database.

---

[3] Erdogan, Ex. 1004, was a printed publication that was publicly available at least as early as January 14, 2008, such that a POSITA, exercising reasonable diligence, could have located it. Moreover, the attached Ex. 1004 is a true and authentic copy as it existed on such date. Ex. 1023.

Erdogan discloses organizing data objects, above called vertices, based upon their relationships to other data objects, above called edges. In Fig. 2, shown below, the "vertices" are the objects at the three different hierarchical levels. The links ("edges") are the paths from one object or vertex to another. Ex. 1003 ¶ 56.

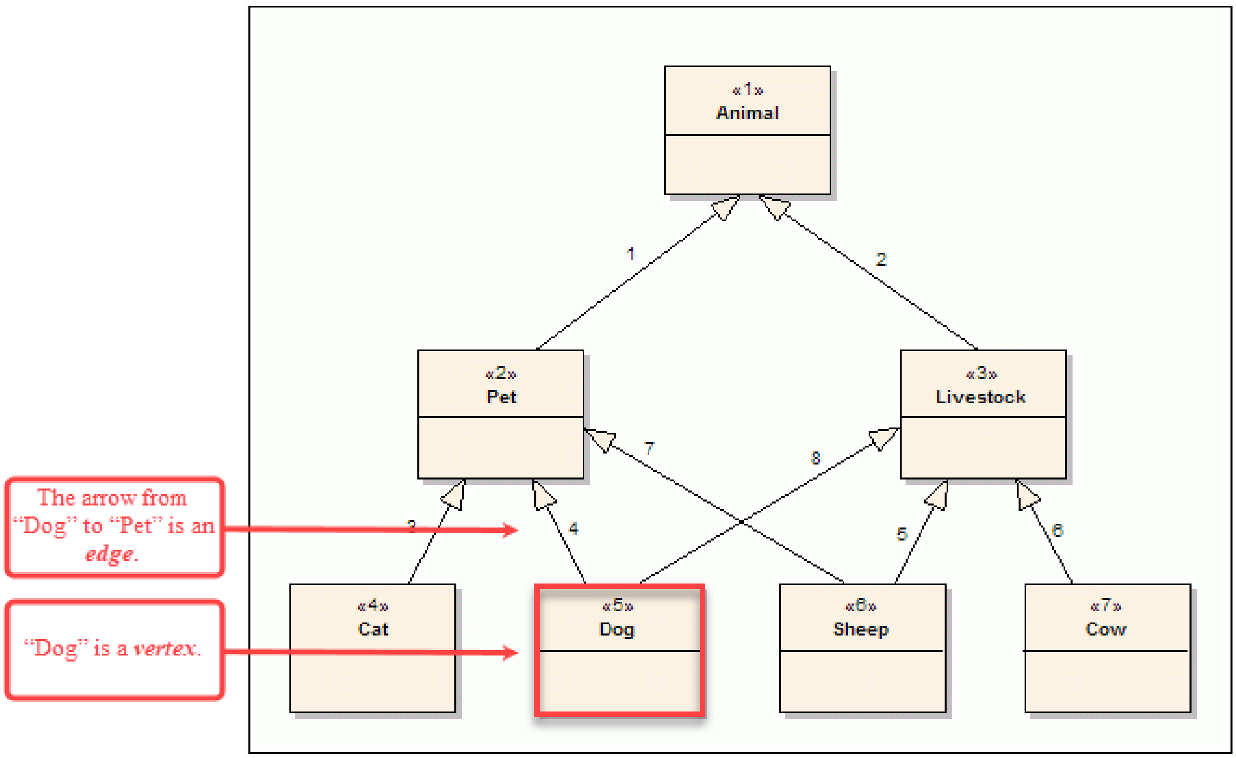


Figure 2: Modified Animal Class Hierarchy

The relationships signified by the arrows above designates the starting object as "is a member of" the ending object. Ex. 1004 at How to Use the Edge Table. For example, the edge connecting "Dog" and "Pet" can be read as "Dog is a member of Pet." The objects and relationships represented above are stored in a helper table called an adjacency list that can be used to update the dataset. This list, shown below

as Table 2, represents the relationships graphically represented above by the number of the starting and ending vertex or object of each edge or relationship. For example, the edge—representing that a Cat, vertex 4, is a "Pet," vertex 2—is edge 3. Ex. 1003 ¶ 57.

Table 2: Representation of Edges on Figure 3 Using an Adjacency List Table

AnimalType Table			Edge Table		
Id	**Name**	**Other Columns**	**EdgeId**	**Start Vertex**	**EndVertex**
1	Animal	...	1	2	1
2	Pet	...	2	3	1
3	Livestock	...	3	4	2
4	Cat	...	4	5	2
5	Dog	...	5	6	3
6	Sheep	...	6	7	3
7	Cow	...	7	6	2
8	Doberman	...	8	5	3
9	Bulldog	...	9	8	5
			10	9	5

Erdogan explains that an object may only have one descriptor in a traditional tree structure. In that case, "Dog" can only be a member of "Pet," and "Sheep" can only be a member of "Livestock." Ex. 1003 ¶ 58.

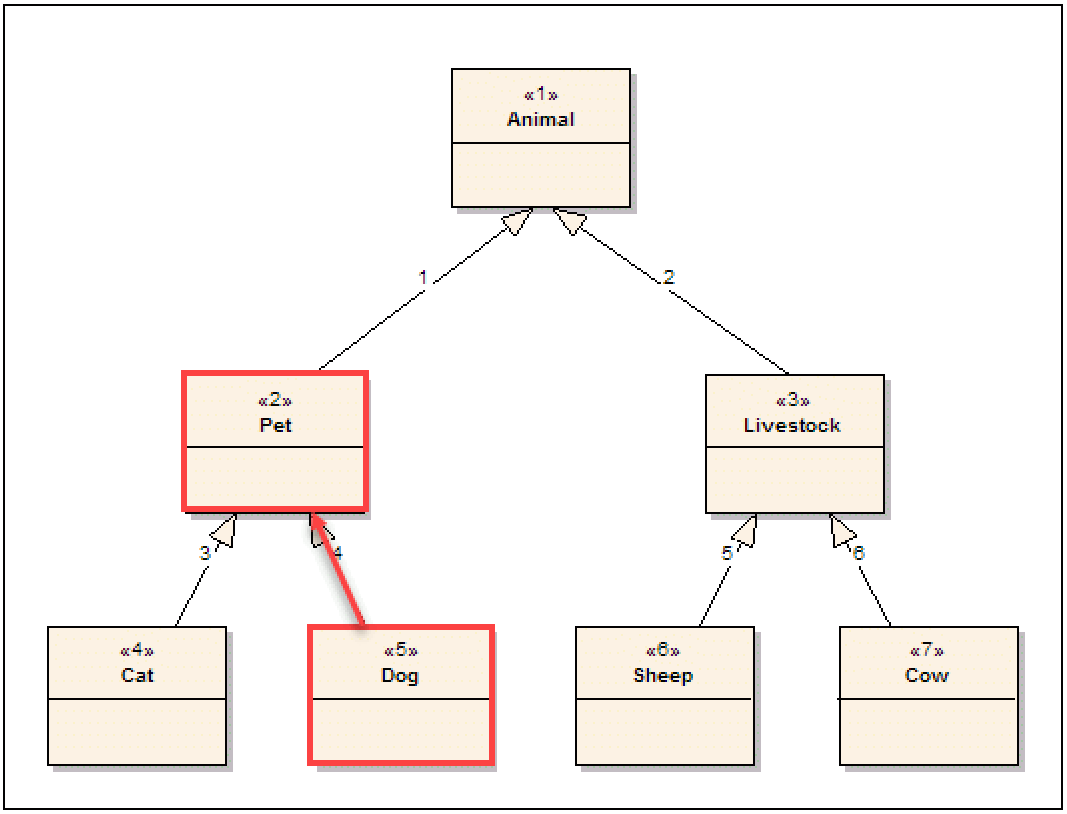


Figure 1: Animal Class Hierarchy

As Erdogan teaches, however, this conventional model is too "restrictive" since in real life, "an object may have many facets, not only one as is suggested by a tree structure." Ex-1004 at The Problem. For example, a lamb could be either livestock or a pet, depending on the circumstance. *Id.* Thus, Erdogan presents the modified class diagram below (Fig. 2) illustrating that objects ***may*** have multiple descriptors in reality. Also, because an object may have multiple descriptors, "now there exist ***multiple paths*** between some vertices." Ex-1004 at The Problem (emphasis added). For example, in Fig. 2, to go from "Dog" to "Animal," there exist

two paths, namely, (Dog → Pet → Animal) and (Dog → Livestock → Animal). Ex. 1003 ¶ 59.

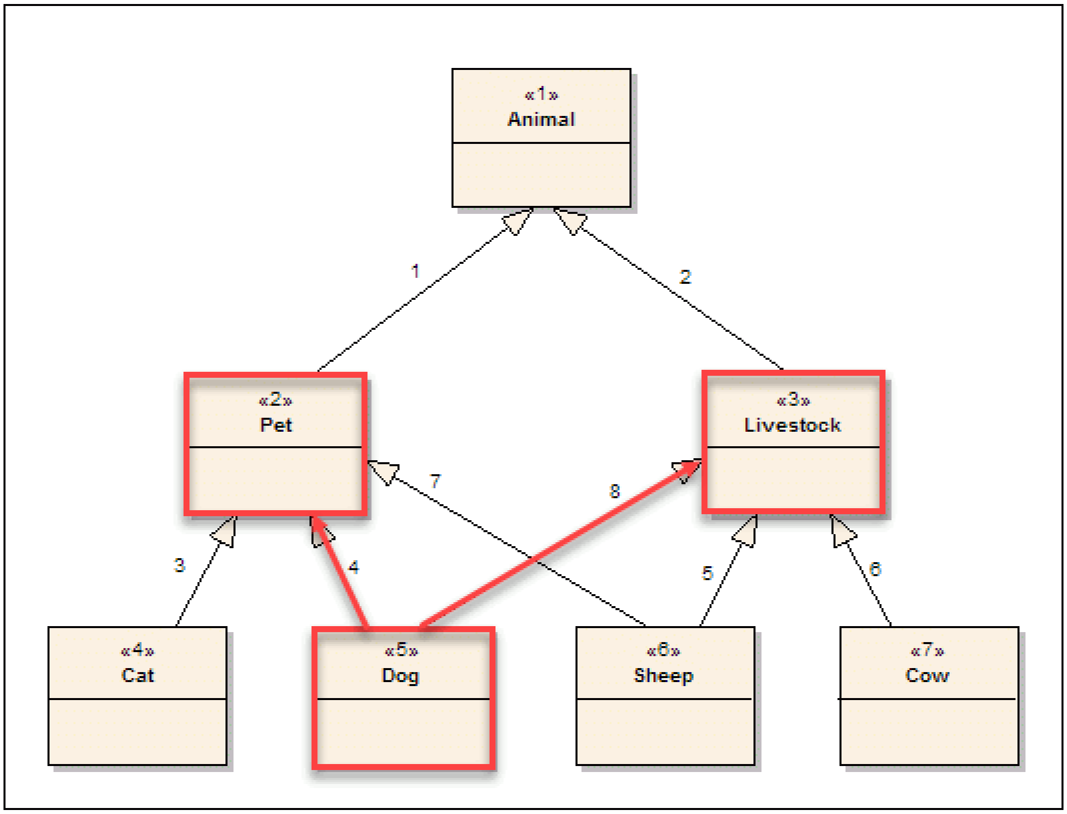


Figure 2: Modified Animal Class Hierarchy

C. Hunt

Hunt, which claims a priority date of March 30, 2005, and published August 11, 2015, describes techniques for analyzing and synthesizing complex knowledge representations (KRs). Both the "multi-level enmeshed directory structure" in the '928 patent and the DAG represented in Erdogan are examples of complex KRs. Ex. 1003 ¶ 61.

According to Hunt, there are many types of KRs, including "taxonomy" and "faceted classification." Taxonomy is a "KR structure that organizes categories into a hierarchical tree and associates categories with relevant objects such as physical items, documents or other digital content." Ex-1005 at 1:54–57. Faceted classification "is based on the principle that information has a multi-dimensional quality" and "can be classified in many different ways." *Id.* at 2:23–25; Ex. 1003 ¶ 62.

Information that has been classified ("concepts") may have two relationship types: (1) intrinsic, referring to joins (edges) between elemental concepts to create more complex concepts—for example, the relationship between "Mountain," "Animal," and "Mountain Animal" in data structure 300 shown in figure below; and (2) extrinsic, referring to joins between complex relationships. *See* Ex-1005 at 11:48–67. Extrinsic relationships may describe features between concept pairs, such as equivalence, hierarchy (for example, the relationship between "Animal" and "Pet"), and associations. *Id.* at 11:53–62; Ex. 1003 ¶ 63.

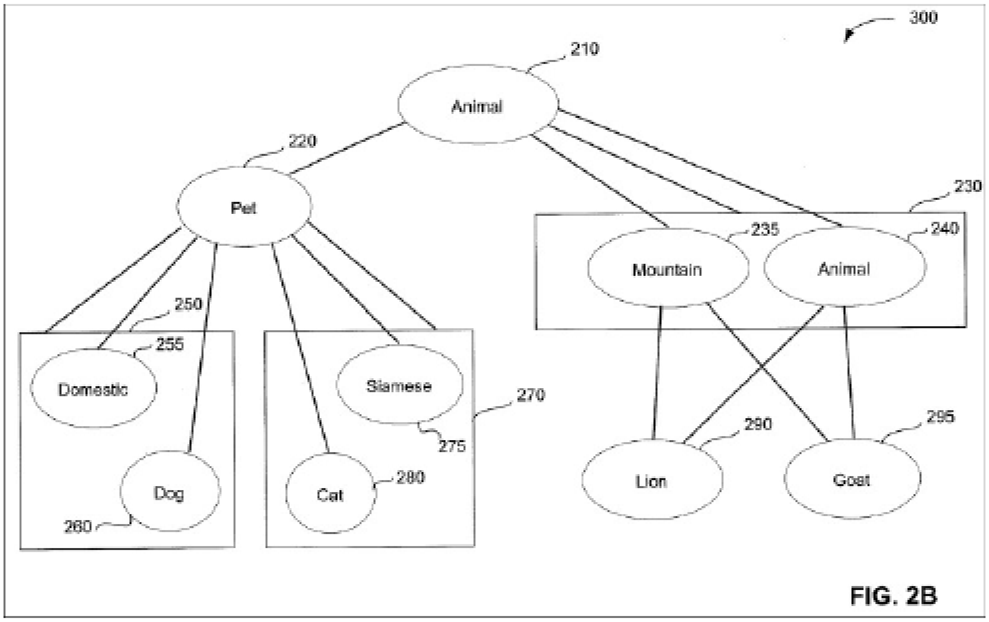


FIG. 2B

Hunt seeks to use "[o]ne or more of the knowledge processing rules" to "analyze an input complex KR to deconstruct its complex concepts and/or concept relationships to elemental concepts and/or concept relationships to be included in the elemental data structure." Ex-1005 at Abstract. Hunt also seeks to use "[o]ne or more of the knowledge processing rules" to "synthesize an output complex KR from the stored elemental data structure in accordance with an input context." *Id.*; Ex. 1003 ¶ 64.

Hunt also describes the presentation of the knowledge representations by a user interface. The "exemplary computer . . . may have one or more input devices and/or output devices, [which] may be used, among other things, to present a user interface." Ex-1005 at 29:1–4; Ex. 1003 ¶ 65.

D. Hekmatpour

Hekmatpour, filed on April 24, 1994 and issued on July 1, 1997, is also directed to structuring data in a multi-level hierarchy. Ex-1006 at Abstract. It is composed of an expert system and processing method employing a three-level hierarchical knowledge base that has a plurality of nodes (vertices) coupled together. *Id.* An uppermost level comprises a behavioral knowledge level, a middle level comprises a structural knowledge level, and a lowermost level is an action level. *Id.*; Ex. 1003 ¶ 67.

One important component of several embodiments of Hekmatpour is the graphical user interfaces (GUI). For example, in the figure below, the user interface is a graphical, window-based interface consisting of visual representations and interaction facilities. Ex-1006 at 21:9–11. Interaction with the system is through a menu-driven, point and click interface, and manual data entry and typing are kept to a minimum. *Id.* at 21:11–13; Ex. 1003 ¶ 68.

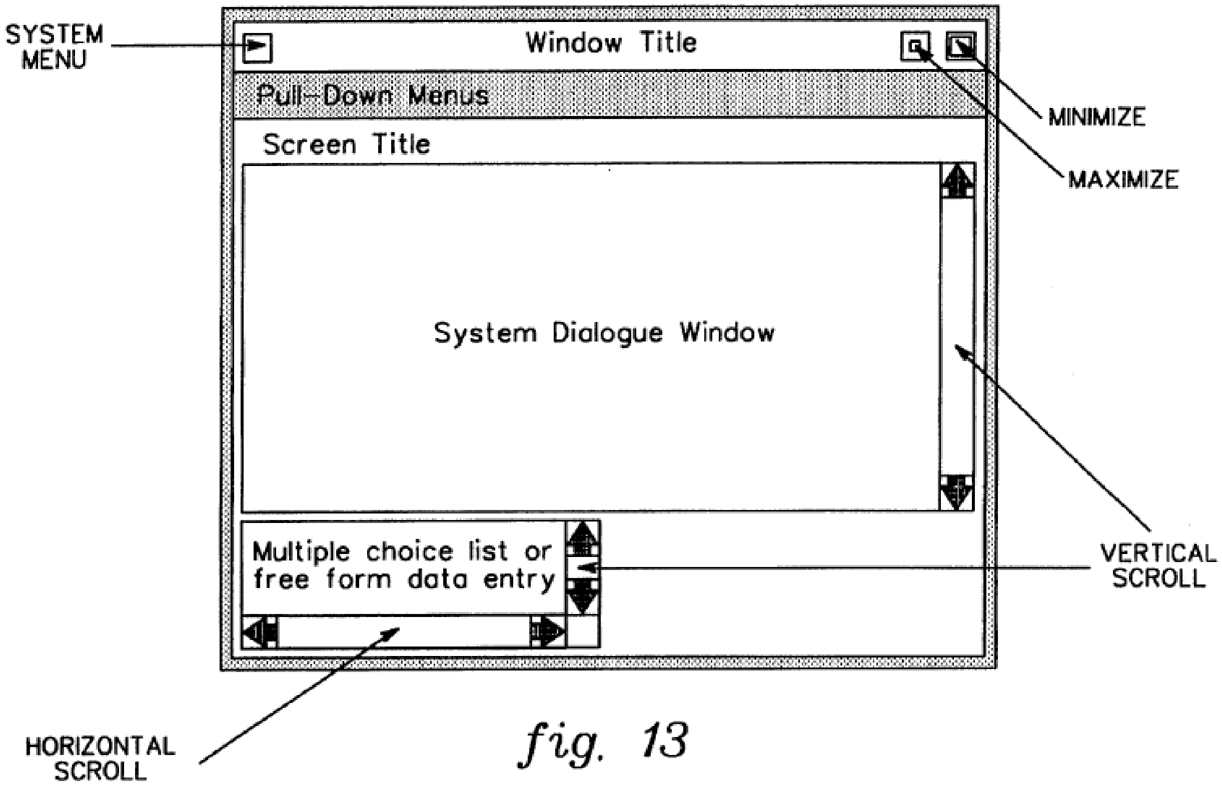


fig. 13

E. Evans

Evans was filed on April 22, 2005, first published on October 26, 2006, and issue on June 5, 2012. Evans, a patent assigned to Microsoft Corporation, relates to systems, methods, and user interfaces that provide navigational tools for storage systems of computers, their operating systems, networks, and the like. Ex-1007 at 1:31–34. The navigational tools in Evans "may provide useful ways of organizing and/or displaying information regarding the user's files, e.g., by hierarchical properties, lists, auto lists, folders, etc." *Id.* at 1:51–53. For example, the figure below shows an example of a user interface for saving a new item (e.g., a file) with

associated hierarchical properties in accordance with examples of this invention. *Id.* at 2:18–20, Fig. 3; Ex. 1003 ¶ 70.

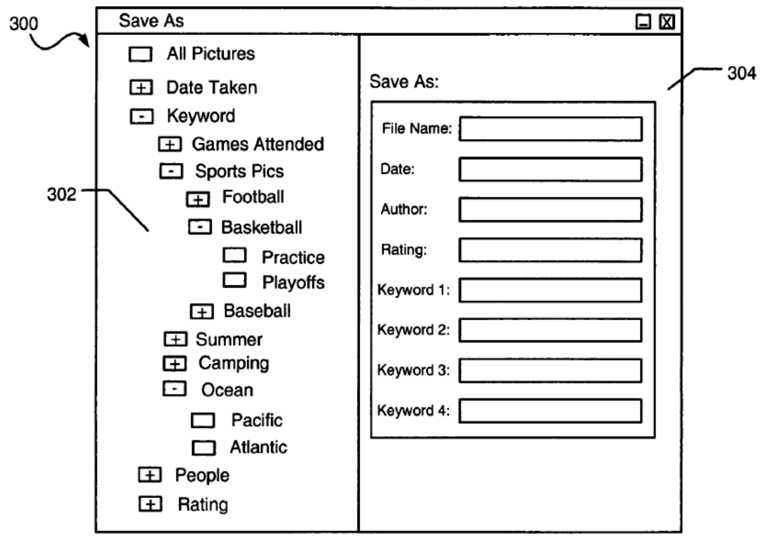


FIG. 3

Evans also discusses Microsoft Corporation's NT File System ("NTFS"). Ex-1007 at 21:12. Under NTFS, the ability to support hard links was introduced to enable users to place electronic files in multiple folders. *Id.* at 21:13. Similar to NTFS, Evans introduces a new end-user concept called a "list." *Id.* at 21:26–27. As a physical analogy, one may think of a "list as a container that references sets of items (e.g., electronic files)." *Id.* at 21:27–29. To better understand "lists," Evans provides a more detailed explanation of a "folder." *Id.* at 21:29–30. A "folder" may be considered as a "set" or group of items that are considered as related to one another in some manner (e.g., being present in the same "folder" may be one way that items in a set may be considered as "related"). *Id.* at 21:31–34; Ex. 1003 ¶ 71.

IX. INSTITUTING THIS *INTER PARTES* REVIEW WOULD BE EQUITABLE

A. *NHK-Fintiv* Factors Support Institution

The Board balances six factors in considering denial under 35 U.S.C. § 314(a). Here, the weight of these factors strongly favors institution. *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) (precedential) ("*Fintiv*").

1. Overlap Between Issues Raised in the Petition and in the Parallel Proceeding

There will be no overlap between this Proceeding and the Litigation, as Petitioner stipulates that, if this *inter partes* review proceeding is instituted, Petitioner will not pursue in the Litigation the grounds raised here, or that could have been reasonably raised (i.e., any ground that could be raised under §§ 102 or 103 on the basis of prior art patents or printed publications). This waiver is intended to be commensurate in scope with the estoppel provisions of 35 U.S.C. § 315(e)(2).

The PTAB has found this kind of stipulation to "mitigate[] any concerns of duplicative efforts between the District Court and the Board, as well as concerns of potentially conflicting decisions." *Sotera v. Masimo Wireless*, IPR2020-01019, Paper 12 at 19 (PTAB Dec. 1, 2020) (Precedential as to § II.A). As such "this factor weighs strongly in favor of not exercising discretion to deny institution." *Id.*; *accord*

Sand Revolution II, LLC v. Cont'l Intermodal Group–Trucking LLC, IPR2019-01393, Paper 24 at 12 n.5 (PTAB June 16, 2020) (informative).

Moreover, and in addition to this waiver, there are additional issues addressed in this Petition that are not addressed in the Litigation. This Petition challenges six more claims than at issue in the Litigation. This Petition challenges claims 1–8 and 13–19, whereas only claims 1–5 and 13–16 are at issue in the Litigation. There are six more claims at issue than in the District Court, and as such there is significant difference in the issues raised in the Petition and in the parallel proceeding.

2. Likelihood of a Stay

Neither party has requested a stay in the parallel Litigation. Typically, a district court stay of the litigation pending resolution of the PTAB trial allays concerns about inefficiency and duplication of efforts. However, here those concerns are alleviated by the above stipulation, which confirms that the PTAB—and the PTAB alone—will address invalidity under §§ 102 and 103 based on patents and printed publications. This approach ensures that *inter partes* review is a "true alternative" to the district court proceeding. *Sand Revolution II*, IPR2019-01393, Paper 24 at 7.

This factor thus weighs in favor of institution or is at least neutral.

3. Proximity of the Court's Trial Date to a Final Written Decision

The District Court entered a Scheduling Order on November 29, 2021 governing the ten patent disputes filed by WSOU. Ex. 1012. Previously, the District Court ordered that the ten patent disputes be split up into three groups with the first group being scheduled for trial on December 5, 2022. Ex. 1013. Trial for the second and third groups are set for February 6, 2023 and March 6, 2023, respectively. The parties have not yet determined which group the '928 Patent will be a part of nor the trial date for the '928 Patent. The parties are to propose to the District Court, by July 8, 2022, the proposed groupings for trial. Ex. 1013.

Further, litigation regarding this patent has already been delayed, putting into serious doubt the district courts tentative trial dates. While a *Markman* hearing was originally scheduled for all ten patents in suit between WSOU and Salesforce on December 15, the court notified the parties on December 13 that the December 15 hearing would only cover five of ten of the patents, in particular excluding the '928 Patent. Ex. 1014. The court explained it would hold a second *Markman* hearing in January, but this hearing has yet to be scheduled at the filing of this Petition.

Petitioner further notes that even in cases where trial is earlier than an expected final written decision, the Board should seek "to balance considerations such as system efficiency, fairness, and patent quality." *Fintiv*, IPR2020-00019,

Paper 11 at 5 (collecting cases). Here, that balance weighs in favor of institution, as, even if trial occurs before a final written decision, Petitioner has filed a broad *Fintiv* waiver that ensures this forum will be the only one to consider patentability under §§ 102 or 103 on the basis of prior art patents or printed publications.

4. Investment in the Parallel Proceeding

Investment in the parallel proceeding has been minimal. WSOU did not serve the complaint until February 24, 2021. Ex-1015; Ex-1016.

At this point, the District Court has invested virtually no time in assessing the parties' claims and defenses—including invalidity. No motion for preliminary injunction was filed. Other than a motion to dismiss the complaint, the Court has issued no order nor scheduled a hearing for any substantive motions. The District Court will have invested little time in addressing the invalidity of the '928 patent by the time this Board decides whether to institute this Petition.

While the District Court will invest some time in deciding the parties claim construction disputes by the *Markman* hearing set for an undefined time in January, it will likely not invest significantly in deciding invalidity until dispositive motions are filed in September 2022. Ex. 1011; Ex. 1012; Ex. 1013. As such the Court is not expected to invest time in determining invalidity until months after an institution decision on this Petition.

Given these circumstances, this factor favors institution.

5. Whether the Petitioner and the Defendant in the Parallel Proceeding are the Same Party

The parties are the same in this *inter partes* review and in the Litigation. Given the stipulation discussed above, however, this factor is neutral.

6. Other Circumstances, that Impact the Board's Exercise of Discretion, Including the Merits

Other circumstances strongly favor institution. In particular, denying institution here will (1) negate the statutorily provided one-year filing period—because this Petition was filed before closure of the one-year window; (2) encourage forum shopping; and (3) condition the institution of *inter partes* reviews on the timing of an oft-changed trial placeholder. No other parties have sought *inter partes* review of the '928 patent.

In particular, Congress, when enacting the America Invents Act, explained that district court defendants should be permitted more than six months to fully evaluate claims in a parallel litigation—extending the *inter partes* review bar date from six months to one year. *See* 157 Cong. Rec. S5429 (Sept. 8, 2011) (Sen. Kyl) (explaining importance of allowing an accused infringer to seek *inter partes* review in view of estoppel, and concomitant need to extend deadline from six months to one year to afford defendants a reasonable opportunity to identify and understand patent claims before filing an *inter partes* review that may trigger estoppel).

Moreover, the merits of the present Petition are particularly strong and outweigh countervailing considerations of efficiency. Had the Examiner been aware of these references during prosecution, the Examiner would not have issued the ’928 patent. As described above, Erdogan and the combinations based on Erdogan and one of Hunt, Hekmatpour, or Evans provide compelling disclosures that render obvious all claims of the ’928 patent. These considerations weigh in favor of institution. *Illumina, Inc. v. Natera, Inc*, IPR2019-01201, Paper 19 at 8 (PTAB Dec. 18, 2019) (instituting when “the strength of the merits outweighs relatively weaker countervailing considerations of efficiency”).

B. 35 U.S.C. § 325(d) Factors Support Institution

Similarly, the Board should not exercise its discretion under § 325(d). Neither of the two prongs relevant to discretionary denial under § 325(d) are applicable here. *Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6 at 7 (PTAB Feb. 13, 2020).

1. None of the Cited Prior Art Was Cited During Prosecution

First, none of the primary references discussed in this Petition have not been considered by the Patent Office in prosecution. They were not considered by the

examiner, cited in an Information Disclosure Statement (IDS),[4] nor were they cited on the face of the '928 patent. Moreover, there are no *inter partes* review or post-grant proceedings involving the '928 patent to date. *See generally* Ex-1001; Ex-1002.

2. The Cited Prior Art Is Not Cumulative of Cited Art

Second, the art cited in this Petition is not cumulative of the art substantively considered by the Examiner—namely, Liang[5] and Schaepe[6] (both first cited by the examiner in the Non-Final Rejection of 8/23/2011).

Discretionary denial under § 325(d) is thus unwarranted. *Pure Storage, Inc. v. Realtime Data LLC*, IPR2018-00549, Paper 7 at 11 (PTAB July 23, 2018).

X. GROUND 1: ERDOGAN AND EVANS RENDER CLAIMS 1–4 OBVIOUS

The hierarchical relational database of Erdogan applied to the directory structure of Evans renders obvious claims 1–4. This combination would have been the natural and obvious implementation of Erdogan. Ex. 1003 ¶ 72.

---

[4] Patent Applicant did not file an Information Disclosure Statement in connection with the patent application. *See* Ex-1002.

[5] Ex-1008.

[6] Ex-1009.

A POSITA would have been motivated to use the database taught by Erdogan as prior systems were "way too complex" and Erdogan taught a more efficient system. Ex-1004 at Introduction. A POSITA seeking to implement the hierarchical database of Erdogan would have naturally begun with a Microsoft system such as Evans. Beyond being a standard in computer science, Microsoft published the Microsoft White Paper Implementing Row- and Cell-Level Security in Classified Databases Using SQL Server 2005 upon which Erdogan improved. *Id.;* Ex. 1003 ¶ 73.

More directly, Erdogan expressly contemplates that the disclosed system could be readily adapted for use in the NTFS file structure taught by Evans. Erdogan teaches that "in Windows, the NTFS file/folder structure (no, it is really not a tree) and Active Directory user/group hierarchy can be modeled using DAGs." *Id.* at The Problem. In other words, Erdogan expressly directs a POSITA to references such as Evans. Ex. 1003 ¶ 74.

Similarly, as explained in Evans, NTFS introduced hard links, which allow users to place electronic files in multiple folders. Ex-1007 at 21:13 (citing Mark Russionovich, Internal Structure of NTFS4.0, (2001)). Erdogan explains its implementations in the terms of real-life objects and relationships such as "Dog," "Pet," "'Dog' is 'Pet.'" A POSITA would have recognized, based on both the general technical knowledge at the time and based on Erdogan's express teachings,

that these real-life objects and relationships can be easily mapped to files, directories, and directory structures, such as "File A," "Folder B," "'File A' is in 'Folder B.'" Ex. 1003 ¶ 75.

Dr. Schmidt further explains a POSITA would be motivated to use the hard links of Evans in the hierarchy taught by Erdogan. Expert at ¶¶ 50–52. In particular, the use of hard and soft links was a common technique used to follow the "DRY" principle, thereby saving memory space and reducing redundancy. *Id*.

Same field of endeavor: Erdogan is directed at representing and generating hierarchical relationships in relational databases. Ex-1004 at Introduction. Likewise, Evans describes a "[c]omputer system navigation tools provide 'links' to various different files, lists, folders, and/or other storage elements and allow users to organize files, e.g., by ***hierarchical*** properties, lists, auto lists, folders, and the like." Ex-1007 at Abstract (emphasis added). As the '928 patent itself notes, "[a] file system is [just] a method for storing and organizing computer files and the data they contain to make it easy to find and access the data." Ex-1001 at 1:11–13. Moreover, "[m]ore formally, a file system is a special purpose database for the storage, organization, manipulation, and retrieval of data." *Id*. at 1:19–21. In other words, a method for representing hierarchical relationship in databases generally as disclosed in Erdogan necessarily encompasses the same method for representing

hierarchical relationships in file systems or, more narrowly, directories, because both file systems and directories are formally types of databases. Ex. 1003 ¶ 76.

Similar techniques to solve the same problems: Both Erdogan and Evans contemplate using the exact same program to solve the same problem. Both Erdogan and Evans contemplate using Microsoft's NTFS to implement hierarchical structures. Ex. 1004 at The Problem; Ex. 1007 at 21:12–24. Because both references contemplate using the same program to implement similar structures, a POSITA would have found it obvious to combine these two references. Ex. 1003 ¶ 77.

Reasonable expectation of success: A POSITA seeking to implement Erdogan on the directory taught by Evans would have had a reasonable expectation of success. A POSITA would have found it trivial to make such an implementation at least because Erdogan directly provides pseudo-code to implement the hierarchical structure it discloses. *See, e.g.*, Ex. 1004 at The Problem. Moreover, Erdogan is directed at teaching its audience to implement its hierarchical structure at least on Microsoft's NTFS. *Id.* NTFS is the same system described by Evans. Ex. 1007 at 21:12–24. Because Erdogan teaches a POSITA with great specificity how to implement its disclosure on the same system described by Evans, a POSITA would have found it obvious to make such an implementation. Ex. 1003 ¶ 78.

A. Claim 1

1. 1pre

A method for generating a multi-level hierarchical directory structure and establishing relationships between descriptors, the method comprising:

Erdogan discloses "modeling [(generating)] ***hierarchical*** data in relational DB [database] tables." Ex-1004 at Introduction (emphasis added). Erdogan likewise establishes relationships between descriptors by disclosing "class ***hierarchies*** with ***multiple*** inheritance," thereby describing a ***multi-level*** hierarchy. *Id.* at The Problem (emphasis added). Petitioner's expert, Dr. Schmidt, has annotated the below figure from Erdogan to highlight the multi-level hierarchy taught by Erdogan. Ex. 1003 ¶ 79.

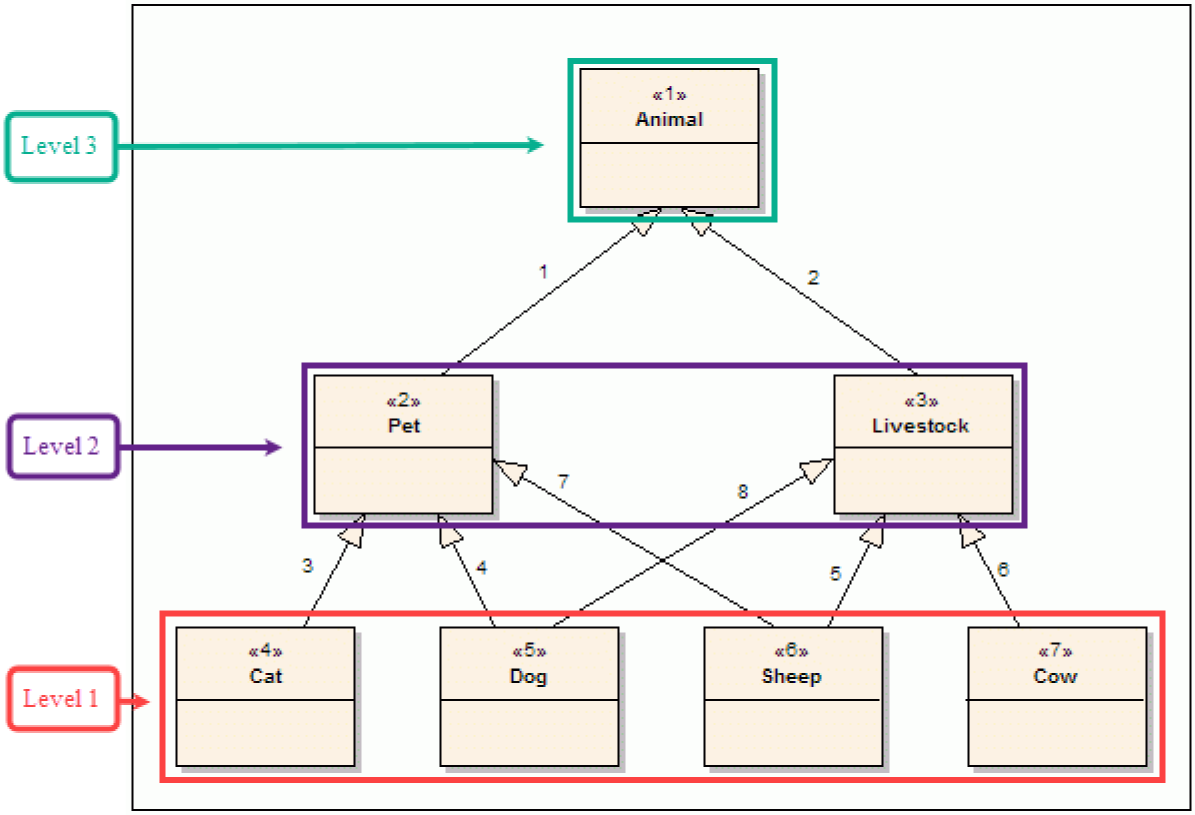


Figure 2: Modified Animal Class Hierarchy

As shown above, Erdogan provides an example of a multi-level hierarchy. In this example, three levels are shown, the top level, Animal, middle level, Pet and Livestock, and bottom level, specific animals. Each level has at least one data object, described by Erdogan as vertices, that are each given a descriptor, for example, Dog, Lamb, or Animal. Each vertex—a data object—is then connected to other vertices, establishing the beginning object "is a" subset of the end object. *See* Ex. 1004 at

Introduction. As such Erdogan teaches "establishing relationships between descriptors" at least during the modeling process.[7] Ex. 1003 ¶ 80.

Erdogan further discloses implementing this hierarchical structure on the claimed "directory structures." Erdogan broadly describes relational databases and specifically contemplates using the Windows NTFS file/folder structure: "in Windows, the NTFS file/folder structure . . . and Active Directory user/group hierarchy . . . can be modeled using" the disclosed system. Ex-1004 at The Problem; Ex. 1003 ¶ 81.

Evans[8] further teaches the use of a directory such as Microsoft's NTFS as a file system that contains a group of files.[9] In particular, as shown in Fig. 3 below, Evans teaches "a user interface for saving a new item (e.g., a file) with associated hierarchical properties." Ex. 1007 at 2:19–21. Evans teaches in such a directory structure a file can appear in multiple folders, similar to Erdogan's teaching that an

---

[7] As explained below in 1c and 1f, by modeling the hierarchy, Erdogan discloses establishing relationships between the descriptors.

[8] While Erdogan suggests a directory structure may be used, it does not provide an in depth discussion of such structures, therefore Petitioner provides Evans for a more detailed discussion of such structures.

[9] As required by the plain meaning of "directory" in the '928 specification.

object can have multiple descriptors. *Id.* at 21:27–29. Evans teaches the use of "hard links" to enable users to place electronic files in multiple folders. *Id.* at 21:13; Ex. 1003 ¶ 82.

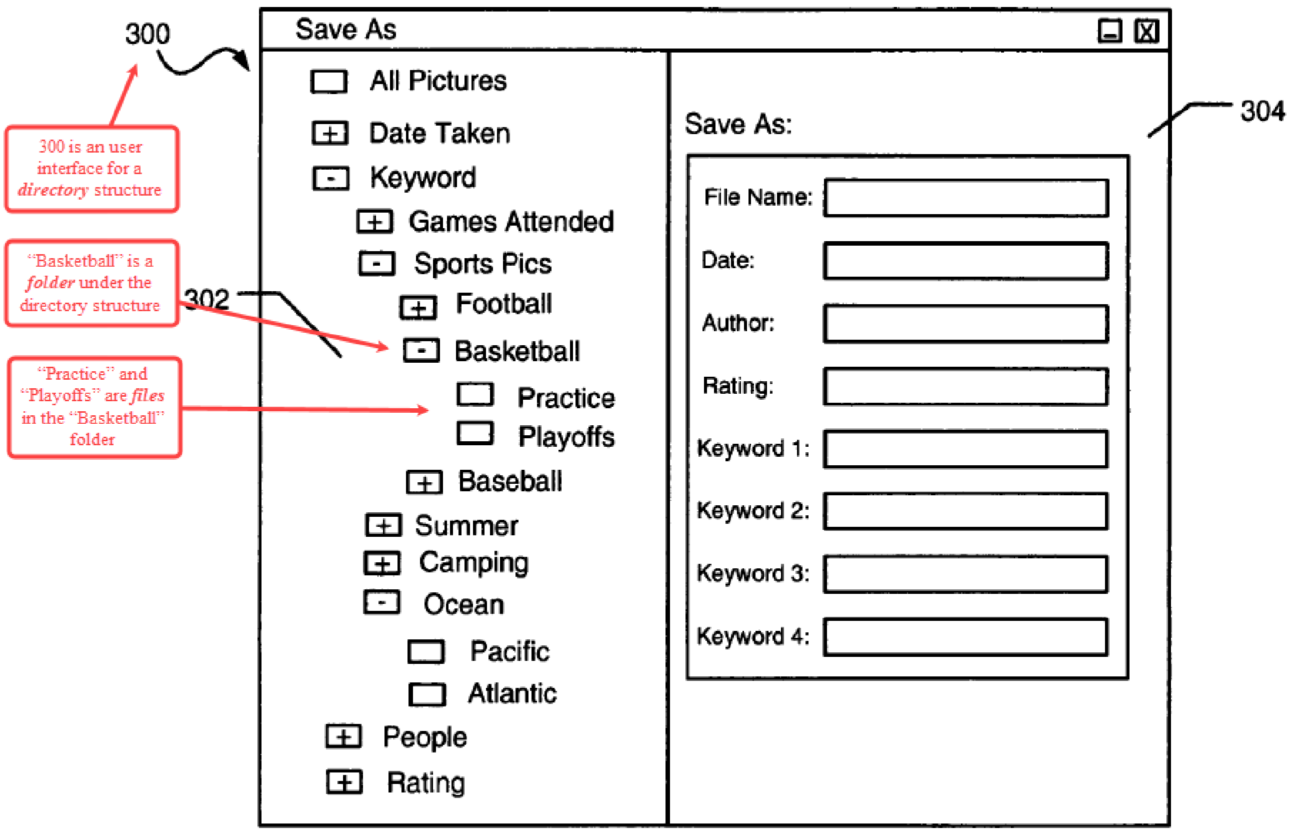


FIG. 3

As such the hierarchical structure taught by Erdogan and implemented on the file structure taught by Evans "generat[es] a multi-level hierarchical directory structure and establish[es] relationships between descriptors." Ex. 1003 ¶ 83.

2. 1a

… selecting an initial data object;

Erdogan discloses a user selecting a starting vertex (initial data object) such as a "Cat" or a "Dog" with the below pseudo code. For example, Erdogan teaches a query operation that expressly requires selecting a start vertex, an initial data object. Erdogan teaches "us[ing] the new structure to get some results. For example, . . . [to] get all the livestock from the tables . . . ." Ex. 1004 at Solution. Erdogan provides example code, annotated below to highlight the select function, for this action wherein an initial object is expressly selected. Ex. 1003 ¶ 84.

Listing 1: SQL Query to Get All Livestock

SQL Copy Code

```sql
SELECT *
     FROM Animal
     WHERE AnimalTypeId IN (
             SELECT StartVertex
                FROM AnimalType
                WHERE EndVertex = 3 )
```

While claim 1 is agnostic about who makes these selections, the '928 Patent contemplates that a human user is making these selections. *See* Ex-1001 at 6:5–8 ("FIG. 4 depicts a flow diagram of a method for generating a multi-level enmeshed directory structure according to one embodiment. The method 400 is entered at step 405 where ***an object of interest is selected***.") (emphasis added). A POSITA would

have understood that only a human user would have an "object of interest." Ex. 1003 ¶ 85. Moreover, WSOU alleges in its Complaint that it is the user who makes the selections: in the Original Complaint for Patent Infringement, WSOU alleges that "Salesforce Sales Cloud allows its users to generate the multi-level hierarchy for any of his or her contacts by selecting contact information (e.g., data object) such as email, mobile, or address." Ex-1022 at 8.

3. 1b

… creating one or more descriptors associated with the data object wherein each of said descriptors are further associated with one or more corresponding descriptors thereby forming a multi-level relational tree;

As explained above with regard to 1pre, Erdogan discusses creating a full hierarchical structure, including "one or more descriptors." *See supra* 1pre. In particular, Erdogan teaches "modeling hierarchical data in relational DB tables." *Id.*[10] Ex. 1003 ¶ 86.

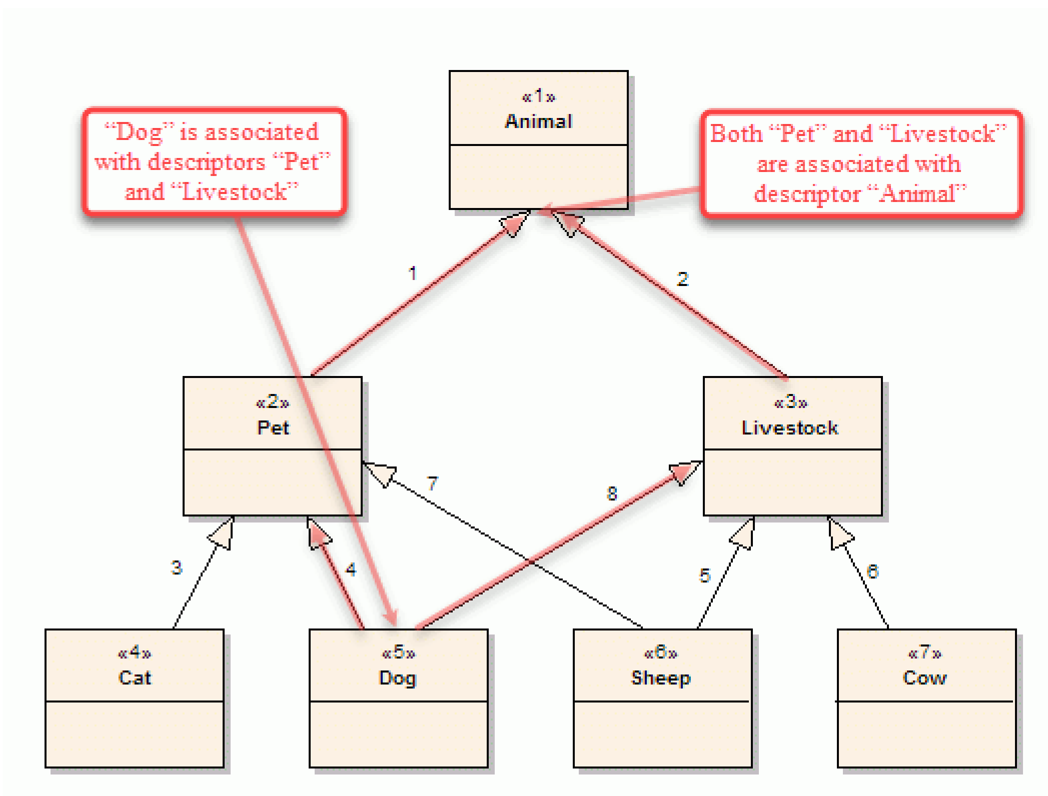
By way of example, in Fig. 2 of Erdogan "Pet" and "Livestock" are "one or more descriptors." Ex. 1004 at Fig. 2. Erdogan explains these descriptors, "Pet"

---

[10] To the extent that Patent Owner argues "modeling" here does not include "creating" a POSITA would have found it obvious to create these additional descriptors. Ex. 1003 ¶ 86.

and "Livestock," are associated with an initial data object "Dog." *Id.* at The Problem. Ex. 1003 ¶ 87.

Erdogan further discloses associating the one or more descriptors with one or more corresponding descriptors. As shown in the example of the annotated figure below, both "Pet" and "Livestock" are likewise associated with descriptor "Animal," thus forming a multi-level relational tree. Ex-1004 at The Problem, Fig. 2; Ex. 1003 ¶ 88.



Similar to 1a, claim 1 is agnostic as to what or who performs this limitation, but the '928 patent contemplates that a human user is again the actor. *See* Ex-1001 at 6:5–8 ("At step 420, ***the object of interest*** is associated with any ***appropriate***

descriptors at a next hierarchical level.") (emphasis added). A POSITA would have understood only a human user would have an "object of interest." Ex. 1003 ¶ 89. And, as above, WSOU alleged in its complaint that the user performs this limitation: "The user can provide descriptors for contact information in the form of contact name and 'reports to' field. The 'reports to' field enable a user to add the contact name of Manager whom the user reports to. This associates a contact name to another contact name (e.g.[,] reporting manager)." Ex-1014 at 8.

4. 1c

… determining the relationship between the one or more descriptors;

The claim does not give much detail as to what "relationship" between the descriptors must be determined. The specification, however, does explain that "the relationship between the one or more descriptors" "can be defined as the set of objects that are described by all of the descriptors in this set." Ex. 1001 at 4:53–55. Consistent with this disclosure, Erdogan discloses determining the relationship between "Pet" and "Livestock." Ex. 1003 ¶ 90.

Erdogan explains that "Dog" is both a type of "Pet" and "Livestock," or that "Sheep" is both a type of "Pet" and "Livestock." Ex-1004 at The Problem, Fig. 2 ("In reality, an object may have many facets . . . . For example, a closer look . . . reveal that, arguably, for some nations dogs are actually livestock, too. Moreover, someone might have a lamb as his/her pet. Figure 2 shows a modified (and still

incomplete) class diagram that ***incorporates these facts***." (emphasis added)). In the same way, Erdogan teaches that both Pet and Livestock are types of Animals. *Id.* (indicating that there is a "path" from "Pet" to "Animal" and another "path" from "Livestock" to "Animal"). As such "the set of objects that are described by all the descriptors" in the set of "the one or more descriptors" is "Dog," "Livestock," and "Pet." Ex. 1003 ¶ 91.

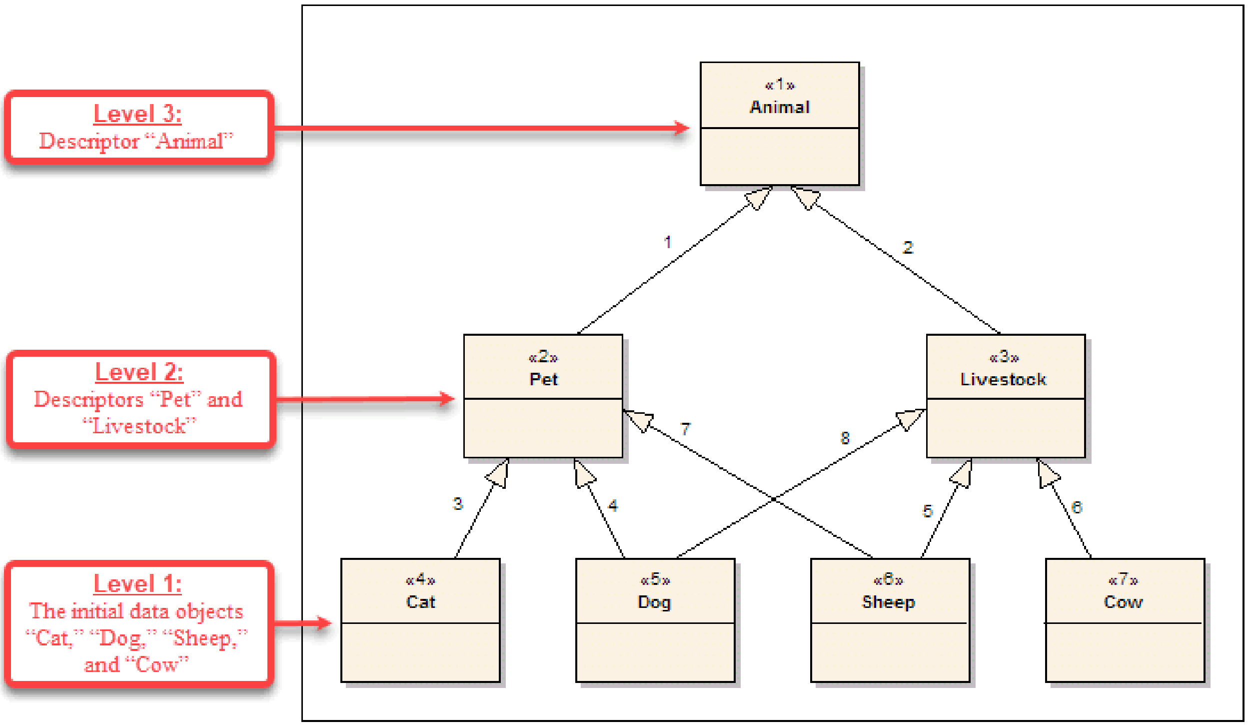
Claim 1 is agnostic about who makes these determinations, but a POSITA would understand that a person can make these determinations. *See* Ex-1001 at 2:65–3:2 ("***User interface device 110*** is a server or any computer including a processor, memory and input/output circuitry that executes at the processor software stored in the memory to implement thereby the Multi-Level Enmeshed Directory Structures system as described herein.") (emphasis added). Ex. 1003 ¶ 92.

5. 1d

… creating a hierarchical structure linking the different levels of descriptors;

As explained *supra* in 1pre, Erdogan discloses creating a multi-level hierarchical structure linking different levels of descriptors. For example, in the figure below, a hierarchical structure for describing the taxonomy for various animals is depicted. Erdogan describes various levels of nesting classifications and

different levels of descriptors as shown in the following annotated figure. Ex. 1003 ¶ 93.



Erdogan discloses "modeling [(creating)] hierarchical data in relational DB tables" such as that shown in the figure above. The initial data objects "Cat," "Dog," "Sheep," and "Cow" on the first level are each linked with one or more descriptors "Pet" and "Livestock" on the second level, which are then linked to descriptor "Animal" on the third level. Ex-1004 at The Problem, Fig. 2. Thus, a corresponding multi-level hierarchy is thus created. *Id.*; Ex. 1003 ¶ 94.

6. 1e

… updating a corresponding database; and

Erdogan discloses updating a corresponding database called an adjacency list. Erdogan defines this adjacency list as "a helper table that stores the edges (the links between vertices) in addition to the table that actually stores" the hierarchical structure described above. Ex. 1004 at The Solution. The adjacency list represents the links between the vertices of the hierarchical structure and is additional to the main table. This matches the disclosure of a "corresponding database" in the '928 patent as simply storing the hierarchical database structure. Ex. 1001 at 6:41–44 ("[T]he database storing the multi-level hierarchical database structure is updated accordingly"). It is therefore a corresponding database. Ex. 1003 ¶ 95.

Erdogan further discloses updating the adjacency list in certain circumstances. For example, Erdogan explains that whenever a new edge is created or an existing edge is deleted, the adjacency list table is updated to reflect the addition or deletion. Ex. 1004 at Insertion of a New edge to the Graph (explaining "[t]he AddEdge procedure creates . . . many . . . redundant implied rows" that "support deletion scenarios"). Ex. 1003 ¶ 96.

Erdogan further notes that additional descriptors (vertices or nodes) can be introduced without disturbing the rest of the graph. *See id.* at The Solution. While Erdogan does not directly disclose updating the adjacency list when adding descriptors, it would have been obvious to a POSITA to make an update in this

situation. Such an update would have been necessary to reflect any updated edges introduced with new vertices. Ex. 1003 ¶ 96.

Therefore, Erdogan discloses that the adjacency list is updated at least whenever an insertion or deletion occurs. Ex. 1003 ¶ 97.

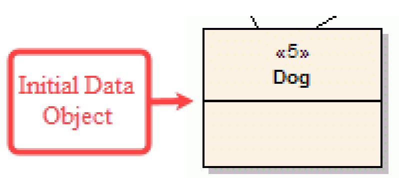
7. 1f

… identifying a single initial descriptor that links a plurality of descriptors and two or more predecessor descriptors linking another single descriptor to thereby establish relationships between different descriptors relative to themselves and to the single initial descriptor.

Erdogan renders this limitation obvious under either party's constructions. Erdogan clearly discloses an initial descriptor, a starting vertex, linked to a plurality of descriptors, further vertices, and predecessor descriptors, called "ancestors" by Erdogan, link to another single descriptor, a final vertex. Moreover, Erdogan teaches organizing these descriptors in a four-level hierarchy as required by the Petitioner's construction. Ex. 1003 ¶ 98.

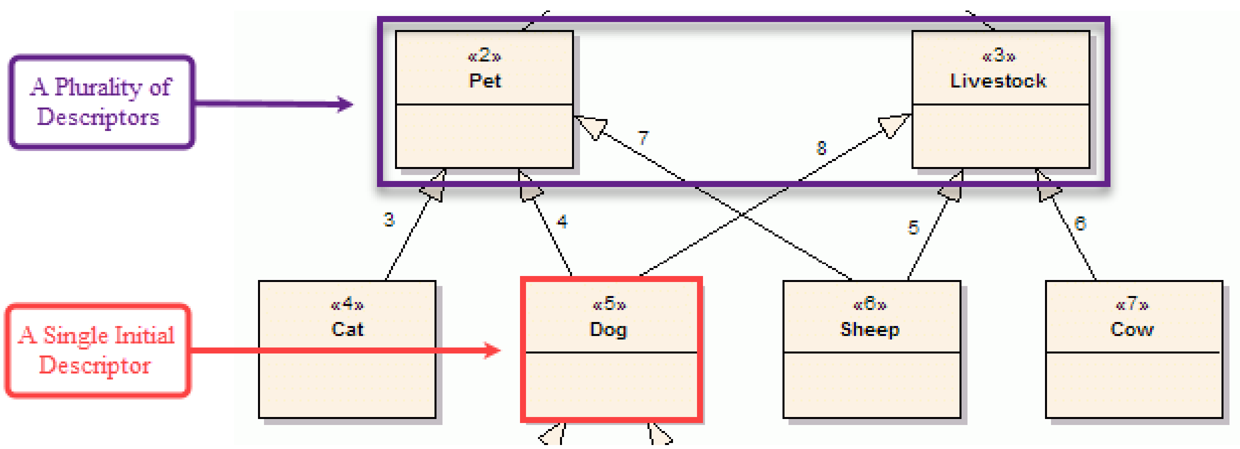
"a single initial descriptor"

Erdogan describes identifying an initial data object, single initial descriptor. *See supra* 1b. For example, "Dog" is a single initial descriptor as shown below. Ex. 1003 ¶ 99.



"that links a plurality of descriptors"

Moreover, Erdogan links the single initial descriptor "Dog" to a plurality of descriptors. As shown in the annotated Figure 3 below, Erdogan links "Dog" as "a single initial descriptor" to both "Pet" and "Livestock" as "a plurality of descriptors." Ex. 1003 ¶ 100.



"***two or more predecessor descriptors"***

Erdogan likewise identifies predecessor descriptors, higher level descriptors describing the plurality of descriptors, called "ancestors." Ex. 1004 at The Problem. For example, in Figure 3, "Pet" and "Livestock" are predecessor descriptors to both Doberman and Bulldog. [11] Thus, Erdogan discloses two or more predecessor descriptors generally. However, Erdogan does not disclose two or more predecessor

[11] These two or more predecessor descriptors are further linked to another single descriptor, "Animal."

descriptors in the dog example above.[12] It would be obvious to a POSITA to add predecessor descriptors to "Dog." Ex. 1003 ¶ 101.

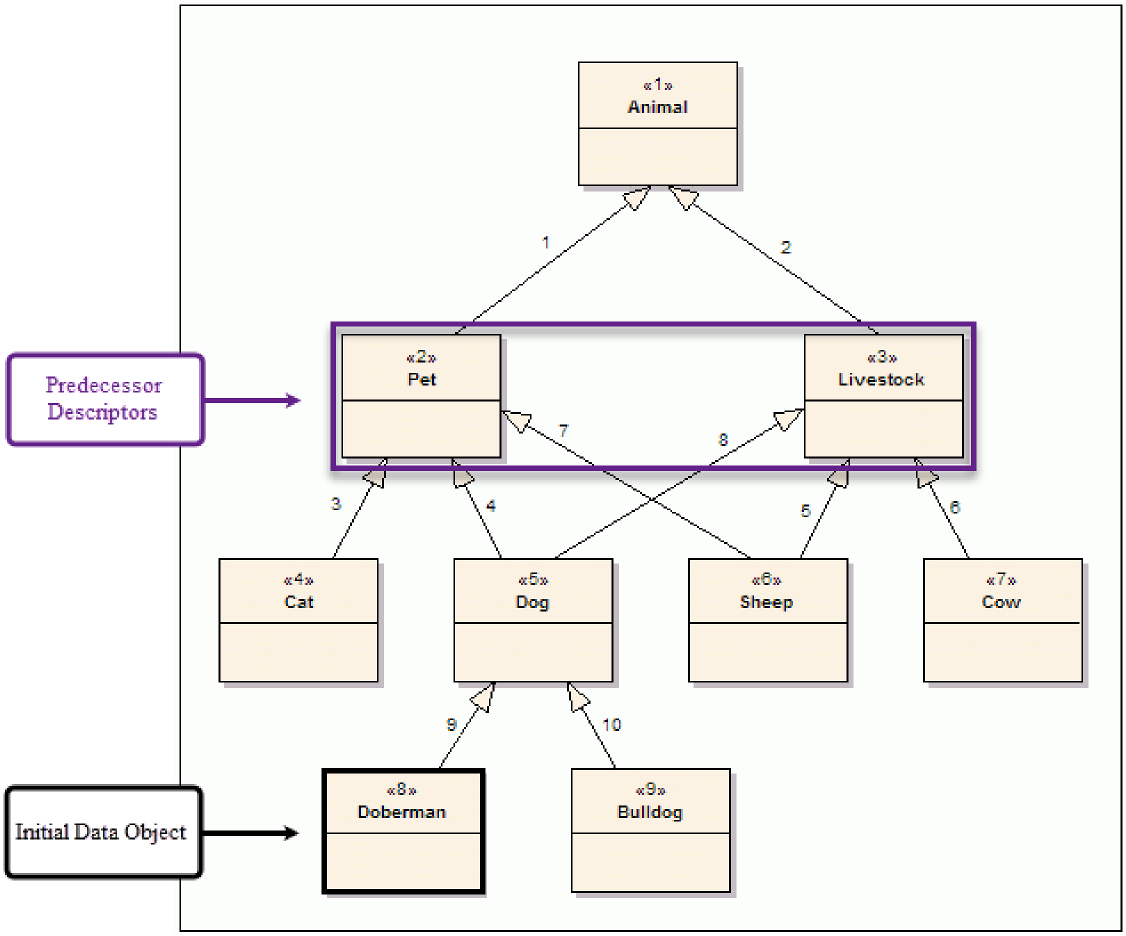


Figure 3: Modified Animal Class Hierarchy

[12] While Erdogan discloses all four levels regarding the Doberman/Bulldog example, it does not disclose the "***plurality*** of descriptors" limitation. Petitioner has therefore chosen to show this ground with "Dog," although Doberman or Bulldog would have been an equally effective illustration.

In particular, a POSITA would have found it obvious to add levels to the dog example, as such an addition is expressly described in the text and contemplated by the very nature of Erdogan (Figure 3). As explained *supra* in 1e, Erdogan provides detailed explanations for insertions and deletions of an edge in a graph. *See* Ex-1004 at The Solution. In particular, additional descriptors (vertices or nodes) can be introduced without disturbing the rest of the graph. *See id.* at The Solution. As explicitly stated in Erdogan, a POSITA can "construct the graph in any order," for example, "you can first create the edge Cat → Pet and then create Pet → Animal" without any disturbance to the rest of the graph. *Id.* at The Solution. In other words, it would be trivial both to add new edges to the example in Fig. 3, such as Pet → Four-Legged and Livestock → Four-Legged, or create an entirely new structure with whatever layers are desired. As such Petitioner's Expert has provided "Four-legged" and "Mammal" as illustrative predecessor descriptors in the figure below. Ex. 1003 ¶ 102.

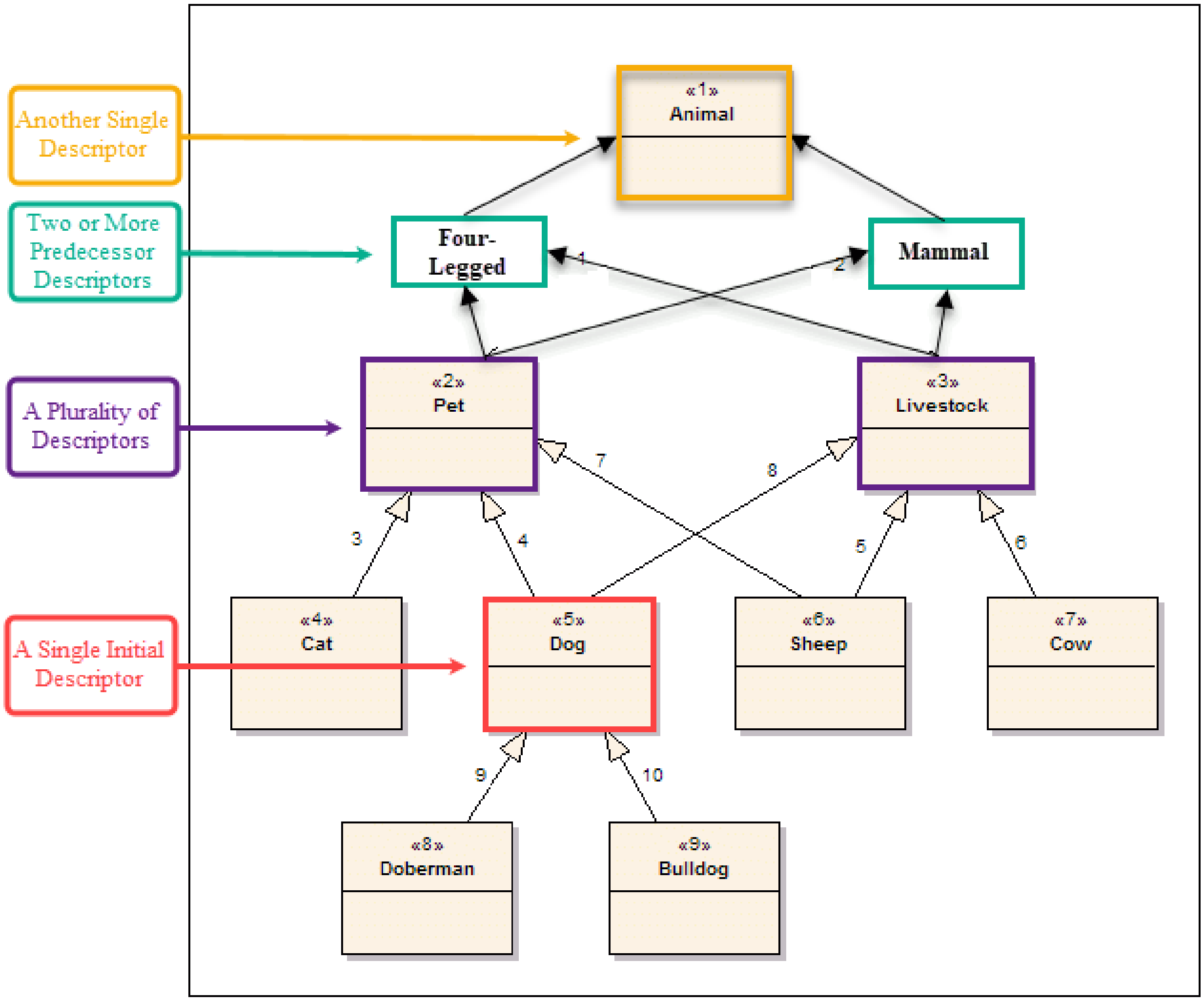


Figure 3: Modified Animal Class Hierarchy

A POSITA would have been motivated to add additional predecessor descriptors as called for by the data set being structured. As Dr. Schmidt explains, multi-level hierarchical structures are used to closely represent the real world. This helps human users to more quickly understand and better manage the representative directory. In this way, using more than a single predecessor descriptor or additional

hierarchical layers generally increases the robustness of the model, thereby allowing it to represent more types of data sets. Ex. 1003 ¶ 103.

"linking another single descriptor"

As shown in the annotated Figure 3 above, Erdogan discloses that two or more predecessor descriptors, "Four-legged" and "Mammal," are further linked to another single descriptor, "Animal." Ex. 1003 ¶ 104.

"establish relationships between different descriptors relative to themselves and to the single initial descriptor"

As explained *supra* in 1c, the '928 Patent explains such a relationship "can be defined as the set of objects that are described by all of the descriptors in this set." Ex. 1001 at 4:53–55. Erdogan discloses establishing "relationships between different descriptors relative to themselves and to the single initial descriptor" modelling the full graph as shown in Figure 3. Ex. 1003 ¶ 105.

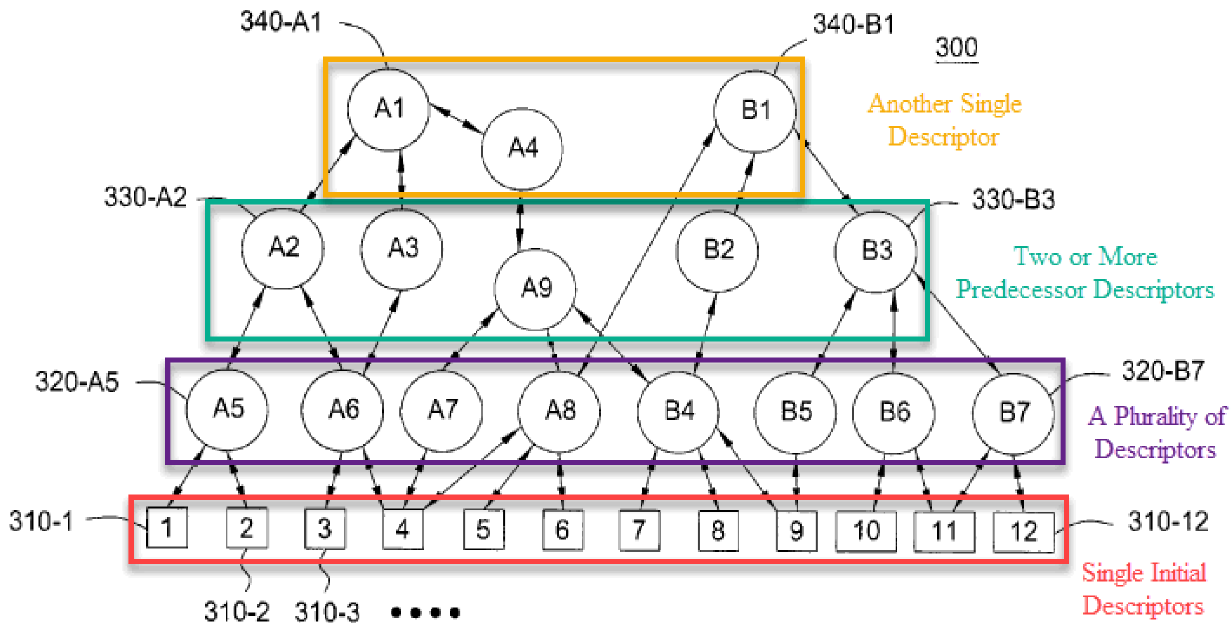
Erdogan discloses "modeling [(establishing)] hierarchical data" such as Figure 3. It would be obvious to a POSITA that "modeling" the hierarchical data in the manner provided by Erdogan "establishes" "the set of objects that are described by all the descriptors in this set." In this case, the set of objects described by all descriptors in the set would be all the vertices shown in the annotated Figure 3. Ex. 1003 ¶ 106.

Moreover, Erdogan discloses the four-level hierarchy in Petitioner's construction of this term. Ex. 1003 ¶ 107.

Four-Level Hierarchy

Erdogan further discloses this limitation according to Salesforce's construction. Erdogan describes "a single initial descriptor ***in a first level of the hierarchy***, the single initial descriptor linked to a plurality of descriptors ***in a second level of the hierarchy***, the plurality of descriptors linked to two or more predecessor descriptors ***in a third level of the hierarchy***, and the two or more predecessor descriptors linked to another single initial descriptor that is ***in a fourth level of the hierarchy***." Ex. 1003 ¶ 108.

Specifically, in the '928 patent, as shown in the annotated figure below, "***a single initial descriptor***" (denoted as 310-1 to 310-12) is in a first level of the hierarchy, which is linked to "***a plurality of descriptors***" (denoted as A5–A8, B4–B7) in a second level of the hierarchy; the plurality of descriptors are then linked to "***two or more predecessor descriptors***" (denoted as A2, A3, A9, B2, B3) in a third level of the hierarchy, which are then linked to "***another single initial descriptor***" (denoted as A1, A4, B1) that is in a fourth level of the hierarchy. Petitioner's expert has annotated the figure used to explain just this limitation during prosecution history. Ex. 1003 ¶ 109.



Erdogan discloses this multi-level hierarchical structure that may comprise four different types of descriptors as explained above. The "***single initial descriptor***" is the first level, for example "Dog." The "***plurality of descriptors***" is the second level, including "Pet" and "Livestock." Next, the "***two or more predecessor descriptors***" are the third level, shown with "Four legged" and "Mammal." Finally, the "***another single initial descriptor***" is the fourth level with "Animal." Petitioner has annotated Figure 3 to highlight these levels below. Ex. 1003 ¶ 110.

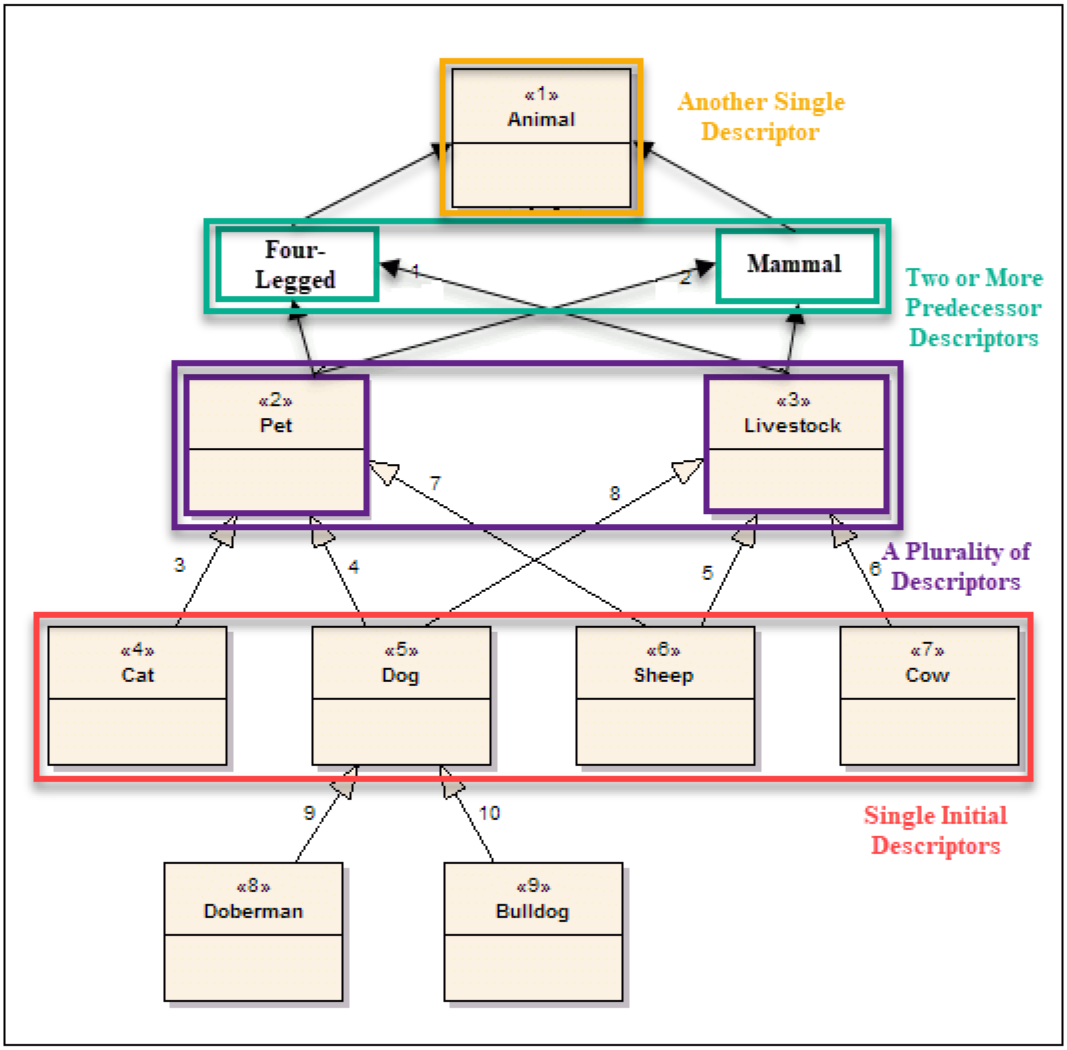


Figure 3: Modified Animal Class Hierarchy

B. Claim 2

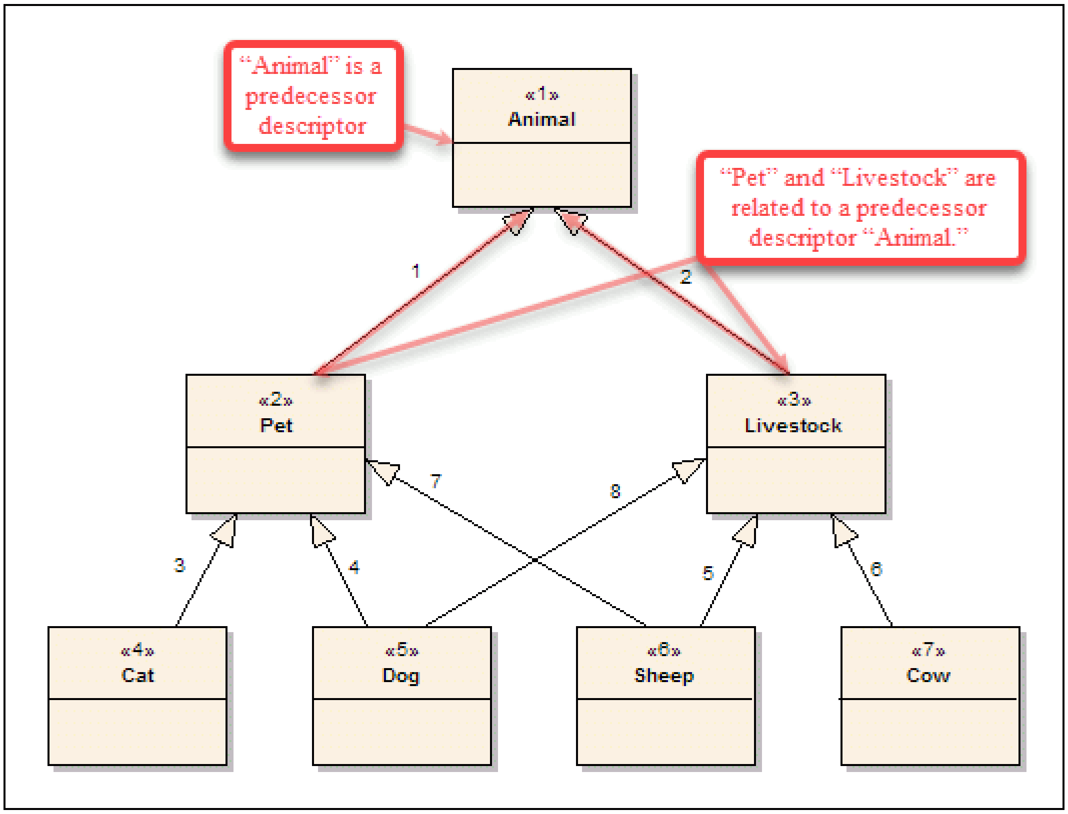
The method of claim 1, wherein each descriptor can be related to one or more predecessor descriptors thereby forming a hierarchical relationship.

Erdogan discloses each descriptor is related to one or more predecessor descriptors, which comprise a superset of descriptors.[13] As shown in 1f, each descriptor, e.g., "Dog" or "Pet," is related to one or more predecessor descriptors, e.g., "Animal." Ex. 1003 ¶ 111.

More specifically, Erdogan discloses the relationship between each descriptor and its one or more predecessor descriptors, thereby forming a hierarchical relationship. Ex-1004 at The Problem. As shown in the annotated figure below, "Pet" and "Livestock" are related to a predecessor descriptor "Animal." *Id.* "Animal" is a predecessor descriptor, or simply a predecessor, because "everything described by [Pet or Livestock] is also described by the predecessor," i.e., "Animal." That is, a "Dog" is a "Pet," and a "Dog" is also an "Animal" (a predecessor of "Dog"). Ex. 1003 ¶ 112.

---

[13] "The intermediate nodes in the graph are descriptors, with ***the predecessor of each descriptor being a superset of the descriptor***. The edge/link between the descriptor and its predecessor is denoted as a "proper link," indicating that the descriptor is a "proper subset" of its predecessor such that everything described by this descriptor is also described by the predecessor." Ex-1001 at 5:1–8 (emphasis added).



C. Claim 3

The method of claim 1, wherein each descriptor can be associated with one or more predecessor descriptors and the relationship of the object to the one or more predecessor descriptors is acyclic.

As explained in *supra* claim 2, Erdogan discloses each descriptor can be associated with one or more predecessor descriptors. Moreover, Erdogan is primarily directed at acyclic structures and as such discloses that the relationship of the object to the one or more predecessor descriptors is acyclic. Ex. 1003 ¶ 113.

In particular, Erdogan is titled "A Model to Represent Directed ***Acyclic*** Graphs (DAG) on SQL Databases." Ex. 1004 (emphasis added). Erdogan explains that "[s]tarting from an arbitrary vertex, if we traverse the vertices in the direction

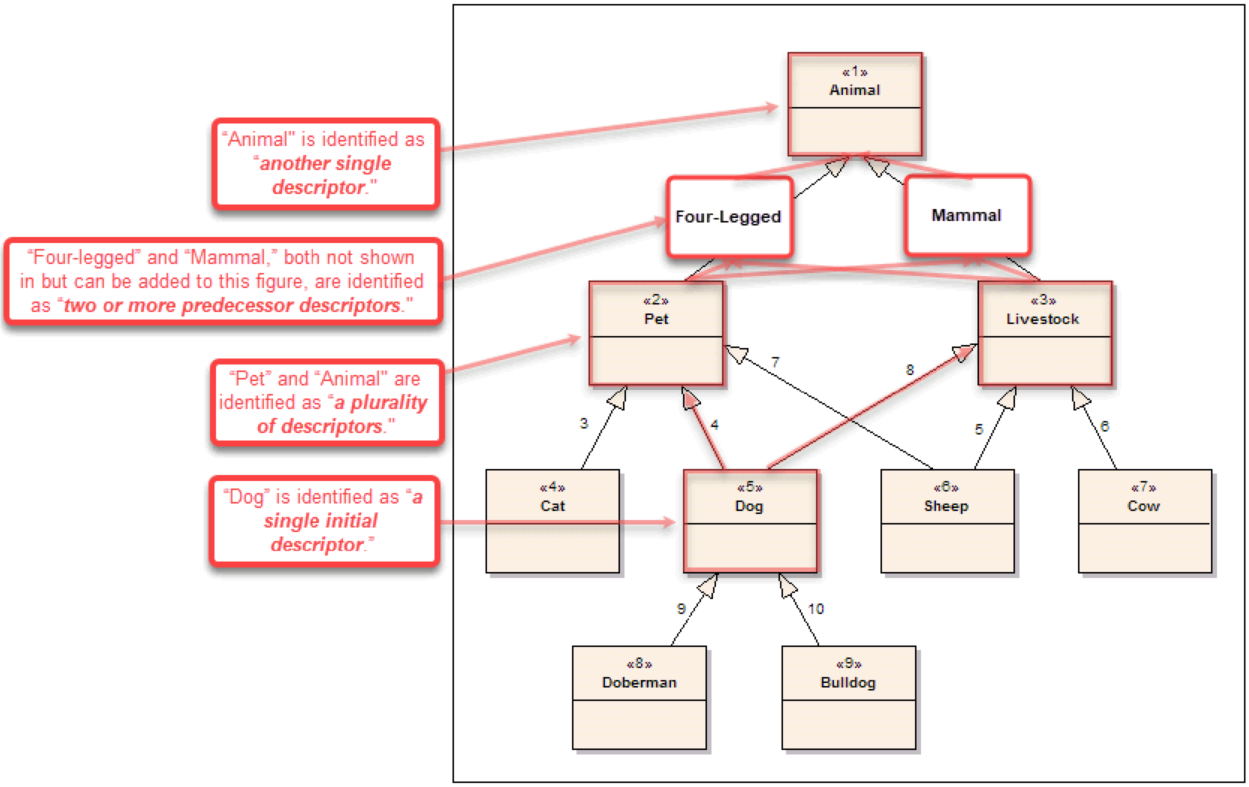
of the arrows, it is not possible to ***return back*** to the originating vertex." *Id*. at Introduction (emphasis added). This terminology is consistent with the way the '928 patent discusses "acyclicity." "Acyclicity" as defined in the '928 patent means that "[t]he resultant enmeshed graph is an . . . graph with all paths ending at an object." Ex. 1001 at 5:1–2; Ex. 1003 ¶ 114.

In the example used in claim 1 above, there exist multiple paths between some vertices (e.g., there are two paths now to go from Dog to Animal), but "it is still not possible to return back to Dog once we leave the Dog vertex." Ex. 1004 at The Problem. Therefore, "it still conforms to the definition of DAG"—that is, the graph is still acyclic. *Id.*; Ex. 1003 ¶ 115.

D. Claim 4

The method of claim 1, further comprising determining the relationships between different descriptors relative to themselves and to the initial data object.

See supra claim 1; Ex. 1003 ¶ 116. As explained above Erdogan discloses determining the relationships between the different descriptors relative to themselves and to the initial data object. *See supra* 1f; Ex. 1003 ¶ 117.



As explained in the annotated figure above, a determination is made that "Dog" is a type of "Pet" and "Livestock." In addition, a determination is made that "Pet" and "Livestock" are both types of "Animal" (as well as "Four-legged" and "Mammal"). Ex. 1004 at The Problem, Fig. 2. A determination is also made that the initial data object (e.g., a particular dog such as a "Doberman" or a "Bulldog") is, for example, a Dog, a Pet, Four-legged, a Mammal, and an Animal. Ex. 1003 ¶ 118.

XI. GROUND 2: ERDOGAN, EVANS, AND HEKMATPOUR RENDER CLAIM 5 OBVIOUS

This ground relies on the disclosures in Erdogan and Evans for the limitations of base claim 1. Erdogan, however, does not expressly disclose claim

5's graphical user interface (GUI). The various elements of the claimed GUI, however, are expressly disclosed by Hekmatpour. Specifically, Hekmatpour discloses GUI tools for a multi-level enmeshed data structure. As discussed below, a POSITA would have combined the GUI elements from Hekmatpour into Erdogan's system at least to improve user experience. Ex. 1003 ¶ 119.

A POSITA would have been motivated to modify the representation of the DAGs in SQL databases taught by Erdogan to further comprise a GUI as taught in Hekmatpour. In particular, this approach would improve user experience by enabling "powerful possibilities for user interface design." Ex. 1006 at 20:67–21:1. The modification aligns with Erdogan's stated purpose of modeling the "more generalized case of Directed Acyclic Graphs" as represented in reality, which naturally entails a user interface to navigate the database. Ex-1004 at The Solution. It also aligns with a longstanding trend as the migration from non-graphical-based operating systems (such as DOS) to graphical-based operating systems (such as Windows). Ex. 1003 ¶ 120.

Same field of endeavor: Erdogan, Evans, and Hekmatpour are in the same field of endeavor as the '928 patent, i.e., representing and generating hierarchical relationships in computer file systems, databases, expert systems, or knowledge representations. *See* Ex-1001 at 1:58–2:2; Ex-1004 at Introduction; Ex-1006 at Abstract; Ex-1007 at Abstract; Ex. 1003 ¶ 121.

Similar techniques to solve the same problems: Erdogan, Evans, and Hekmatpour also use similar techniques to solve the same problems as the ’928 patent. All three references involve the same techniques as the ’928 patent to use multiple object descriptors to describe a single object, despite having slightly different terminologies. Ex. 1003 ¶ 122.

Reasonable expectation of success: Market forces would have favored combining Erdogan and Hekmatpour. “Many researchers in the field of human-computer interaction and cognitive science have studied the effects and requirements of such an interface environment on users as well as system design.” Ex. 1006 at 21:1–5. Market pressure therefore existed to generate GUIs for methods or systems of modeling hierarchical data in relational databases. The increased demand would prompt variations of user interfaces and provide design incentives and market forces that would have motivated a POSITA to combine references, including Erdogan, Evans, and Hekmatpour, to achieve automated representation of database data in a customizable and user-friendly manner. Ex. 1003 ¶ 123.

Moreover, a POSITA would have had a reasonable expectation of success in combining Erdogan, Evans, and Hekmatpour because they disclose straightforward techniques that are based on standard database technology and could be readily implemented by a competent programmer. Ex. 1003 ¶ 124.

A. Claim 5

The method of claim 1, wherein the operation further comprises a graphical user interface (GUI) to navigate the enmeshed directory in both directions, said GUI presents both descriptors or objects described by a particular descriptor using proper links and the descriptors describing the object or descriptor using proper links.

Hekmatpour expressly discloses embodiments that use a GUI in the field of hierarchical data structures in multi-level enmeshed structure. Applying this GUI to the teachings of Erdogan would present both descriptors by a particular descriptor using proper links and the descriptors describing all descriptors using proper links as shown in Figure 3. Ex. 1003 ¶ 125.

Hekmatpour expressly discloses the use of a GUI for structuring data in a multi-level hierarchy. Ex-1006 at 4:52–56, cl 13. Thus, claim 5 is obvious in light of Hekmatpour's disclosure that a method for creating a multi-level enmeshed structure can involve a GUI to navigate the enmeshed directory in both directions. Ex. 1003 ¶ 126.

In particular, Hekmatpour explains its "user interface methodology is based on IBM's Common User Access (CUA) architecture." Ex. 1006 at 21:13–15. "The CUA user interface is a graphical interface that incorporates elements of object-orientation, i.e., an orientation in which the user's focus in on objects and in which the concept of applications is hidden." *Id.* at 21:49–51. Object orientation facilitates presenting objects by their descriptors and is the same methodology considered by

Erdogan. Ex. 1004 at Introduction ("[A] notable and familiar structure . . . is the inheritance hierarchy in Object-oriented Programming."). Hekmatpour presents an example of this GUI with regard to a training example. *Id.* at 28:27–36. As can be seen in the annotated Fig. 21 below Hekmatpour's "GUI presents both descriptors or objects described by a particular descriptor using proper links and the descriptors describing the object or descriptor using proper links." Ex. 1003 ¶ 127.

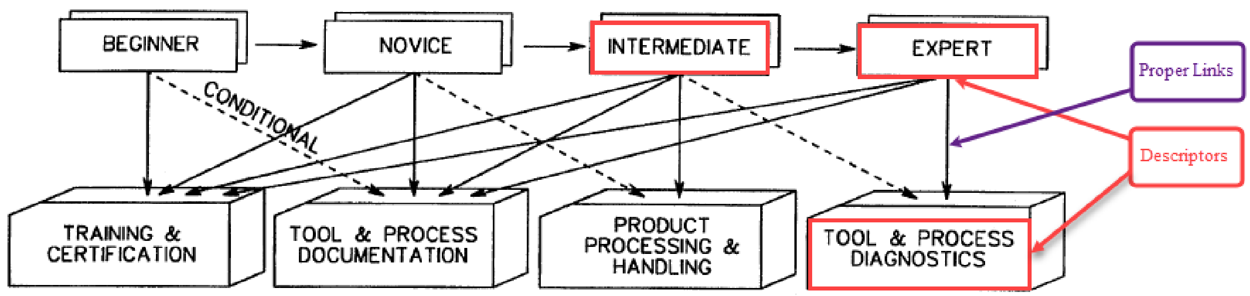


fig. 21

To the extent Patent Owner argues that Hekmatpour's Fig. 21 does not present descriptors described by other descriptors are required by claim 5, it would have been obvious to a POSITA to apply the GUI of Hekmatpour to the descriptors describing further descriptors of Erdogan. Ex. 1003 ¶ 128.

XII. GROUND 3: ERDOGAN, EVANS, HEKMATPOUR, AND HUNT RENDER CLAIMS 6–8, 13-19 OBVIOUS

This ground similarly relies on two base references, Erdogan and Evans, to provide disclosure of modeling and implementation of DAGs in both SQL databases and in the context of directory structures. Further, it relies on a third reference, Hunt, to provide disclosures of Erdogan's techniques in the context of "***a network***

management system" (claims 6–8), "***a computer readable storage medium***" (claims 13–16), and "***a content server***" (claims 17–19). This ground similarly relies on Hekmatpour as described in Ground 2 above. Ex. 1003 ¶ 129.

It would have been obvious to a POSITA to implement the of DAGs in SQL databases taught by Erdogan on "a network management system" (claims 6–8), "a computer readable storage medium" (claims 13–16), and "a content server" (claims 17–19) as taught by Hunt. Ex. 1003 ¶ 130.

Erdogan itself contemplates its system can be applied to other circumstances. For example, Erdogan explains there are many implementations of the DAGs taught including "in Windows, the NTFS file/folder structure . . . and Active Directory user/group hierarchy . . . can be modeled using DAGs." *Id.* This explanation would motivate a POSITA to apply the teachings of Erdogan in other areas already using hierarchical data. Ex. 1003 ¶ 131.

In particular, a POSITA would have been motivated to combine Erdogan, Evans, and Hunt for the following reasons. Ex. 1003 ¶ 132.

Same field of endeavor: Erdogan, Hunt, Hekmatpour, and Evans are in the same field of endeavor as the '928 patent, i.e., representing and generating hierarchical relationships in computer file systems, databases, expert systems, or knowledge representations. *See* Ex-1001 at 1:58–2:2; Ex-1004 at Introduction; Ex-1005 at 1:54–57; Ex-1007 at Abstract. Evans, like the '928 patent, is directed at the

file system directly. In addition, the ’928 patent itself notes that “[a] file system is [just] a method for storing and organizing computer files and the data they contain to make it easy to find and access the data.” Ex-1001 at 1:11–13. Moreover, “[m]ore formally, a file system is a special purpose database for the storage, organization, manipulation, and retrieval of data.” *Id.* at 1:19–21. In other words, a method for representing hierarchical relationship in databases generally as disclosed in Erdogan necessarily encompasses the same method for representing hierarchical relationship in file systems or, more narrowly, directories, because both file systems and directories are formally all types of databases. Ex. 1003 ¶ 133.

Similarly, Hunt, which is directed at knowledge representations (KRs) is a broader extension of the concept disclosed in the ’928 patent. For example, as noted earlier, both the “multi-level enmeshed directory structure” in the ’928 patent and the DAGs (directed acyclic graphs) represented in Erdogan are examples of complex KRs. Ex. 1003 ¶ 134.

Similar techniques to solve the same problems: Erdogan, Hunt, Hekmatpour, and Evans also use similar techniques to solve the same problems as the ’928 patent. All three references involve the same techniques as the ’928 patent to use multiple object descriptors to describe a single object, despite having slightly different terminologies. For example, instead of using objects, descriptors, and links to

describe a hierarchical relationship, Erdogan uses objects (vertices), facets (vertices), and paths (edges) to do so. Ex. 1003 ¶ 135.

Both Erdogan and Evans reference Microsoft Corporation's NT File System ("NTFS"). Ex-1004 at The Problem; Ex-1007 at 21:12. Under NTFS, the ability to support hard links was introduced. This feature enabled users to place electronic files in multiple folders. Ex-1007 at 21:13; Ex. 1003 ¶ 136.

Reasonable expectation of success: As with Erdogan and Evans, a POSITA seeking to implement Erdogan on any variety of environments would have had a reasonable expectation of success. A POSITA would have found it trivial to make such an implementation at least because Erdogan directly provides pseudo-code to implement the hierarchical structure it discloses. *See, e.g.*, Ex. 1004 at The Problem. Further, Erdogan is directed at teaching its audience to implement its hierarchical structure at least on an SQL Server. *Id.* This is very similar to the various networking and server environments described by Hunt. Because Erdogan teaches a POSITA with great specificity how to implement its disclosure on the similar systems to those of Hunt, it would have been obvious to make such an implementation. Ex. 1003 ¶ 137.

A. Claims 6–8

Claim	Corresponding Claim Limitation
6pre	1pre
6a	n/a, see discussion *infra*

6b	1b–1d
6c	1f
7	5
8	3

The table above lists the relationship between claims 6–8 and claims 1–5. Petitioner does not argue that these corresponding claims use identical language, but rather that they are rendered obvious by the same prior art for the same reasons discussed above. For example, 6pre is similar to 1pre, but is directed at "[a] network management system communicatively coupled to one or more element management systems." Ex. 1001 at 11:14–15. Limitations (except for 6a) and dependent claims of claim 6 can all find their respective equivalent in claims 1–5. Ex. 1003 ¶ 138.

1. 6pre

A network management system communicatively coupled to one or more element management systems adapted to perform a method for creating a multi-level hierarchical directory structure and establishing relationships between descriptors, comprising:

Hunt discloses a network management system. Hunt explains a computer for the purpose of analyzing and synthesizing complex knowledge representations, which includes any hierarchical relationship, "may also comprise one or more network interfaces . . . to enable communication via various networks," examples of which include a local area network or a wide area network, such as an enterprise network or the Internet." Ex-1005 at 29:13–18. Such networks may be based on any suitable technology and may operate according to any suitable protocol and may

include wireless networks, wired networks or fiber optic networks. *Id.* at 29:18–21. As such a POSITA would have been motivated and enabled to combine Erdogan with the network management system described in Hunt. Ex. 1003 ¶ 139.

2. 6a

… a processor for executing software instructions received from a memory to perform thereby a method for, the method comprising:

While a POSITA would have found it obvious to include "a processor for executing software instructions received from a memory" in most any computer science system, claim 1 of Hunt, discloses a "computer-implemented method for generating a complex knowledge representation," which comprises "a processor, in accordance with the request context." Ex-1005 at 32:17–18, 32:20–21; Ex. 1003 ¶ 140.

3. 6b

… linking each of a plurality of data objects to multiple respective descriptors, each of said descriptors being linked with one or more predecessor tags; and

See claim limitations 1b–1d; Ex. 1003 ¶ 141.

4. 6c

… identifying a single initial descriptor that links a plurality of descriptors and two or more predecessor descriptors linking another single descriptor to thereby establish the relationships between different descriptors relative to themselves and to the single initial descriptor.

See claim limitation 1f; Ex. 1003 ¶ 142.

5. 7

The system of claim 6, wherein the operation further comprises a graphical user interface (GUI) to navigate the enmeshed directory in both directions, said GUI presents both descriptors or objects described by a particular descriptor using proper links and the descriptors describing the object or descriptor using proper links.

See claim 5; Ex. 1003 ¶ 143.

6. 8

The method of claim 6, wherein the relationships of the object to the one or more descriptors is acyclic.

See claim 3; Ex. 1003 ¶ 144.

B. Claims 13–16

Claim	Corresponding Claim Limitation
13pre	1pre
13a	1b–1d
13b	1f
14	5
15	3
16	4

The table above lists the relationship between claims 13–16 and claims 1–5. Petitioner does not argue that these corresponding claims use identical language, but rather that they are rendered obvious by the same prior art for the same reasons discussed above. For example, 13pre is similar to 1pre, but is directed at "[a] computer readable storage medium for storing instructions [that may be] executed by one or more processors communicatively coupled to a network." Ex. 1001 at

12:4–6. Limitations and dependent claims of claim 13 can all find their respective equivalent in claims 1–5. Ex. 1003 ¶ 145.

1. 13pre

A computer readable storage medium for storing instructions which, when executed by one or more processors communicatively coupled to a network, perform a method for creating a multi-level hierarchical directory structure and establishing relationships between descriptors, comprising:

This limitation is rendered obvious by Erdogan in combination with Hunt. One embodiment in Hunt "is directed to a system for generating a complex knowledge representation, the system comprising at least one non-transitory computer-readable storage medium storing processor-executable instructions." Ex-1005 at 3:30–33; Ex. 1003 ¶ 146.

2. 13a

… linking by a device an object to multiple descriptors describing said object, each of said descriptors being identified by one or more predecessor descriptors linked to the descriptor; and

See claim limitations 1b–1d; Ex. 1003 ¶ 147.

3. 13b

… identifying a single initial descriptor that links a plurality of descriptors and two or more predecessor descriptors linking another single descriptor to thereby establish relationships between different descriptors relative to themselves and to the single initial descriptor.

See claim limitation 1f; Ex. 1003 ¶ 148.

4. 14

The computer readable medium of claim 13, wherein navigating the enmeshed directory in both directions requires the GUI, said GUI presents both descriptors or objects described by a particular descriptor using proper links and the descriptors describing the object or descriptor using proper links.

See claim 5; Ex. 1003 ¶ 149.

5. 15

The computer readable medium of claim 13, wherein the relationship of the object to the one or more descriptors is acyclic.

See claim 3; Ex. 1003 ¶ 150.

6. 16

The computer readable medium of claim 13, further comprising determining the relationships between different descriptors relative to themselves and to the initial data object.

See claim 4; Ex. 1003 ¶ 151.

C. Claims 17–19

Claim	Corresponding Claim Limitation
17pre	1pre
17a	1b–1d
17b	1f
18	5
19	3

The table above lists the relationship between claims 17–19 and claims 1–5. Petitioner does not argue that these corresponding claims use identical language, but rather that they are rendered obvious by the same prior art for the same reasons discussed above. For example, 17pre is similar to 1pre, but is directed at "[a] content

server comprising a processor, said content server multicasting to a plurality of client servers in a network system." Ex. 1001 at 12:31–33. Limitations and dependent claims of claim 17 can all find their respective equivalent in claims 1–5. Ex. 1003 ¶ 152.

1. 17pre

A content server comprising a processor, said content server multicasting to a plurality of client servers in a network system adapted to perform a method for creating a multi-level hierarchical directory and establishing relationships between descriptors, the method comprising:

To the extent the preamble is found limiting, Erdogan and Hunt disclose this limitation. Hunt discloses a content server comprising a processor and multicasting to a plurality of client servers on a network. One embodiment disclosed in Hunt entails "a system for analysis and synthesis of complex [knowledge representations]," which "may be implemented on a server side of a distributed computing system with network communication with one or more client devices, machines and/or computers." Ex-1005 at 18:17–22; Ex. 1003 ¶ 153.

While the '928 patent does not specifically define a content server, the claim language explains a content server "compris[es] a processor" and "multicast[s] to a plurality of client servers in a network system." The '928 patent describes a content server as "multicasting to a plurality of content servers where the top level descriptors are the servers serving content," and describes client servers as "the

lowest level objects." Ex-1001 at 7:36–38. As noted above, Hunt teaches a server, the content server, which distributes information to client devices, the lowest level objects. It would have been obvious to a POSITA therefor to implement the SQL server taught by Erdogan as the server taught by Hunt, thereby rendering this limitation obvious. *See supra* 1pre; Ex. 1003 ¶ 154.

2. 17a

… linking by the content server each of a plurality of client servers to multiple respective descriptors, each of said descriptors being linked with one or more predecessor descriptors wherein a top level predecessor descriptor corresponds to the content server; and

See claim limitations 1b–1d; Ex. 1003 ¶ 155.

3. 17b

… identifying a single initial descriptor that links a plurality of client servers and two or more predecessor descriptors linking another single descriptor to thereby establish relationships between different descriptors relative to themselves and to the single initial client server.

See claim limitation 1f; Ex. 1003 ¶ 156.

4. 18

The content server of claim 17, further comprising generating a graphical user interface (GUI) to visualize navigation of the enmeshed directory in both directions, said GUI presenting both descriptors or clients servers described by a particular descriptor using proper links and descriptors describing the client server or descriptor using proper links.

See claim 5; Ex. 1003 ¶ 157.

5. 19

The network system of claim 17, wherein each descriptor can be associated with one or more predecessor descriptors and the relationships of the client server to the one or more descriptors is acyclic.

See claim 3; Ex. 1003 ¶ 158.

XIII. CONCLUSION

For at least the foregoing reasons, this Petition should be instituted.

DATED: January 11, 2022

/s/ James Glass
James Glass (Reg. No. 46729)

CERTIFICATION UNDER 37 C.F.R. § 42.24

Under the provisions of 37 C.F.R. § 42.24, the undersigned hereby certifies that the word count for the foregoing Petition for *inter partes* review (excluding the table of contents, table of authorities, mandatory notices, certificate of service or word count, and appendix of exhibits or claim listing) totals 11,798 words, which is within the word limit allowed under 37 C.F.R. § 42.24(a)(i).

Date: January 11, 2022

/s/ James Glass
James Glass (Reg. No. 46729)

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§ 42.6(e), 42.105(a), the undersigned hereby certifies service on the Patent Owner of a copy of this Petition and its respective exhibits at the official correspondence address for the attorney of record for the '928 patent as shown in USPTO PAIR via FedEx:

Etheridge Law Group
2600 Southlake Blvd.
Suite 120-324
Southlake, TX 76092

Courtesy copies were also sent via electronic mail to Patent Owner's counsel of record in the related district court proceeding, Case No. 6:20-cv-01165-ADA (W.D. Tex.) at the following addresses:

James L. Etheridge
Jim@EtheridgeLaw.com

Ryan S. Loveless
Ryan@EtheridgeLaw.com

Travis L. Richins
Travis@EtheridgeLaw.com

Brett A. Mangrum
Brett@EtheridgeLaw.com

Jeffrey Huang
Jhuang@EtheridgeLaw.com

Mark D. Siegmund
mark@waltfairpllc.com

Date: January 11, 2022

/s/ James Glass
James Glass (Reg. No. 46729)